

Universidad Autónoma Metropolitana
Unidad Azcapotzalco
Posgrado en Optimización

**Técnicas metaheurísticas de
optimización multiobjetivo
para resolver el problema del
portafolio de inversión**

Tesis presentada por
Lic. Naim Reyes Hernández

Para obtener el grado de Maestro en Optimización

Asesor
Dr. Antonin Ponsich

08 de Septiembre del 2017



Técnicas metaheurísticas de optimización multiobjetivo para resolver el problema del portafolio de inversión

Esta tesis está dedicada a mi familia, porque a su lado he forjado esos valores que permiten reconocer y estimar las mejores dádivas que nos puede obsequiar la vida, entre muchas de ellas el conocimiento y la ciencia

Contenido

Índice de objetos	IV
Introducción	1
1 Marco teórico	5
1.1 El problema del portafolio de inversión.....	6
1.1.1 El modelo de Markowitz	6
1.1.2 Extensiones: otros modelos	7
1.2 Las instancias.....	8
1.3 Optimización: generalidades.....	10
1.3.1 Optimización monobjetivo	11
1.3.2 El paradigma de los algoritmos genéticos.....	12
1.3.3 Manejo de restricciones	15
1.4 Optimización multiobjetivo	16
1.4.1 Generalidades	17
1.4.2 Taxonomía de los enfoques multiobjetivo	20
1.4.3 Métricas.....	21
2 Técnicas Metaheurísticas en Optimización Multiobjetivo	23
2.1 Algoritmos de selección de Poblaciones No Dominadas (PNDs)	23
2.1.1 El algoritmo exhaustivo (A1)	25
2.1.2 Algoritmo versión II (A2)	25
2.1.3 Clasificador de Bentley (M3)	27
2.1.4 Análisis de eficiencia	28
2.2 Algoritmo Genético basado en una suma ponderada de los objetivos	30
2.2.1 La noción de función de agregación.....	30
2.3 La estrategia Vector Evaluated Genetic Algorithm (VEGA).....	31
2.4 La estrategia Nondominated Sorting Genetic Algorithm II (NSGA-II)	34
2.4.1 Jerarquía de dominancia	35
2.4.2 Crowding distance assignment	38
2.5 Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D).....	40

2.5.1	Boundary Intersection (BI) Approach.....	41
3	Ajuste de parámetros y manejo de restricciones	46
3.1	Métricas utilizadas	46
3.1.1	Distancia generacional	47
3.1.2	Efficient Set Spacing	48
3.1.3	Hipervolumen.....	49
3.2	Detalles de codificación, hardware y software	51
3.3	Metodología para la configuración de parámetros	52
3.3.1	Fase 1: valores extremos.....	53
3.3.2	Fase 2: diseño factorial completo	54
3.3.3	Las decisiones de desempeño en el entorno multiobjetivo.....	56
3.4	Criterio de paro y análisis del tamaño de las instancias	58
3.4.1	Pruebas de alcance y la función para el criterio de paro	63
3.5	Estrategias de selección, cruza y mutación.....	68
3.5.1	Selección.....	69
3.5.2	Cruza y Mutación.....	70
3.5.3	Análisis comparativo	74
3.6	Metodología para el manejo de restricciones	78
3.6.1	Normalización.....	79
3.6.2	Reparación aleatoria	80
3.6.3	Reparación max-min	82
3.6.4	Análisis comparativo	83
3.7	Ajuste de parámetros de las técnicas para el problema.....	91
3.7.1	Suma Ponderada de Objetivos	92
4	Desarrollo experimental.....	103
4.1	Implementación 1: Ajuste temporal	103
4.2	Implementación 2: resultados finales.....	110
4.3	Análisis de decisión bajo el criterio Hipervolumen	114
4.3.1	Eficiencia de las Metaheurísticas	114
4.3.2	Análisis gráfico de los frentes de Pareto aproximados	118
4.4	Análisis empírico global.....	124
4.5	Comparación con el estado del arte	126

Conclusiones	132
Bibliografía	136
Apéndices	138
Interpretación de resultados en las pruebas de hipótesis	138
El problema del tamaño de archivo EP	143
Ajuste de parámetros para las estrategias VEGA, NSGA-II y MOEA/D	148
VEGA.....	148
NSGA-II	153
MOEA/D	163
Estadísticos descriptivos de implementación final tiempo ajustado	176

Índice de objetos

Algoritmo 1.1 Genético monobjetivo (minimizar)	14
Algoritmo 2.1 Comparador de soluciones Bentley (Bentley, Kennet, & David , 1993).....	24
Algoritmo 2.2 Selector de conjunto de Pareto A2	26
Algoritmo 2.3 Selector de conjunto de Pareto M3 (Bentley, Kennet, & David , 1993).....	27
Algoritmo 2.4 VEGA Canónico	33
Algoritmo 2.5 NSGA-II	37
Algoritmo 2.6 Cálculo de Crowding distance	40
Algoritmo 2.7 MOEA/D	44
Algoritmo 3.1 Selección por torneo binario (minimizar)	70
Algoritmo 3.2 Método de reparación aleatoria	81
Algoritmo 3.3 Método de reparación max-min	82
Algoritmo 0.1 propuesta de restricción de archivo EP en la estrategia MOEA/D.....	147
Figura 1.1 Extensiones del modelo de Markowitz (Ponsich, López Jaimes, & Coello, 2013).....	8
Figura 1.2 Frentes de Pareto reales para las instancias estudiadas.....	10
Figura 1.3 Técnicas de optimización (Collette & Siarry, 2004).....	12
Figura 1.4 Algoritmo Genético monobjetivo.....	14
Figura 1.5 Diferencia entre conjunto y frente de Pareto	19
Figura 2.1 Tiempo (s) de ejecución, Llamadas al comparador de soluciones, selectores de PNDs ..	28
Figura 2.2 Gráficas Rojo: Ψ^{2m} Verde: $\Psi[\log_2(\Psi)]m - 1$	29
Figura 2.3 Etapas de un algoritmo basado en la estrategia VEGA	32
Figura 2.4 Etapas de un algoritmo basado en la estrategia NSGA-II.....	34
Figura 2.5 Ejemplo jerarquías de dominancia (minimización)	37
Figura 2.6 Generación de frentes NSGA-II	38
Figura 2.7 Cálculo del Crowding-distance en un espacio de dos objetivos	39
Figura 2.8 Boundary Intersection (BI) Approach, fuente: (Zhang & Li, 2007).....	42
Figura 2.9 Etapas de un algoritmo basado en la estrategia MOEA/D	44
Figura 3.1 Cálculo de la distancia generacional	48
Figura 3.2 Área del hipervolumen en \mathbb{R}^2	50
Figura 3.3 Ejemplo de Análisis factorial completo: dos valores para tres parámetros.....	52
Figura 3.4 Generación de intervalos en ajuste de parámetros.....	54
Figura 3.5 Experimento: decisión de desempeño.....	57
Figura 3.6 Métricas (ordenadas) respecto al tamaño de la instancia (abscisas)	60
Figura 3.7 Frentes aproximados respecto al tamaño de la instancia.....	61

Figura 3.8 Comportamiento de métricas al tamaño de la instancia	62
Figura 3.9 Métricas, prueba de alcance HVperc = 90%	64
Figura 3.10 Llamadas a la función objetivo, prueba de alcance HVperc = 90%	65
Figura 3.11 Estimación de función para el criterio de paro (NSGA-II)	68
Figura 3.12 Función $g(N)$ estimada (verde) y propuesta (rojo)	68
Figura 3.13 Métodos de cruza-mutación (a) todas las estrategias	75
Figura 3.14 Métodos de cruza-mutación (b)	76
Figura 3.15 Métodos de cruza-mutación VEGA	77
Figura 3.16 Mejores resultados por método de cruza-mutación (hipervolumen)	78
Figura 3.17 Métricas por estrategia	84
Figura 3.18 HVperc manejo de restricciones (MOEA/D)	85
Figura 3.19 HVperc manejo de restricciones (NSGA-II)	86
Figura 3.20 HVperc manejo de restricciones (SPO)	87
Figura 3.21 HVperc manejo de restricciones (VEGA)	87
Figura 3.22 Manejo de restricciones: gd vs. HVperc	89
Figura 3.23 Manejo de restricciones: gd vs. ESS	90
Figura 3.24 Manejo de restricciones: ESS vs. HVperc	91
Figura 3.25 Ajuste parámetros etapa1: HVperc (SPO) (a)	93
Figura 3.26 Ajuste parámetros etapa1: tiempo (SPO)	94
Figura 3.27 Ajuste parámetros etapa1: HVperc vs. tiempo (SPO)	94
Figura 3.28 Ajuste parámetros etapa1: HVperc (SPO) (b)	96
Figura 3.29 Ajuste parámetros etapa1: gd , ESS y card (SPO)	96
Figura 3.30 Ajuste parámetros etapa2: tiempo (SPO)	97
Figura 3.31 Ajuste parámetros etapa2: HVperc (SPO) (a)	98
Figura 3.32 Ajuste parámetros etapa2: HVperc (SPO) (b)	99
Figura 3.33 Ajuste parámetros etapa2: HVperc (SPO) (c)	100
Figura 3.34 Ajuste parámetros etapa2: gd , ESS y card (SPO)	101
Figura 4.1 Implementación 1: HVperc	103
Figura 4.2 Implementación 1: tiempo	105
Figura 4.3 Ejemplo ajuste temporal inst0 (segundos)	107
Figura 4.4 Implementación 1: HVperc comparación con ajuste temporal	109
Figura 4.5 Implementación 2: HVperc	112
Figura 4.6 Implementación 2: Distancia Generacional	113
Figura 4.7 Implementación 2: ESS	113
Figura 4.8 Implementación 2: Cardinalidad	114
Figura 4.9 Desempeño de las técnicas (HVperc)	117
Figura 4.10 Inst0 mediana (a)	119
Figura 4.11 Inst0 mediana (b)	119
Figura 4.12 Inst1 mediana (a)	120
Figura 4.13 Inst1 mediana (b)	120
Figura 4.14 Inst2 mediana (a)	121
Figura 4.15 Inst2 mediana (b)	121

Figura 4.16 Inst3 mediana (a).....	122
Figura 4.17 Inst3 mediana (b)	122
Figura 4.18 Inst4 mediana (a).....	123
Figura 4.19 Inst4 mediana (b)	123
Figura 4.20 Estado del arte, Distancia Generacional (gd)	128
Figura 4.21 Comparación con el estado del arte (gd)	130
Figura 0.1 Ejemplo: gráfica de distribuciones normales (a)	140
Figura 0.2 Ejemplo: gráfica de distribuciones normales (b).....	141
Figura 0.3 Ejemplo: gráfica de distribución chi-cuadrada con 3 grados de libertad.....	143
Figura 0.4 Ejecución exhaustiva inst2: HVperc	144
Figura 0.5 Ejecución exhaustiva inst2: cardinalidad	145
Figura 0.6 Ejecución exhaustiva inst2: Frente aproximado	145
Figura 0.7 Ejecución exhaustiva inst2: gd y ESS	146
Figura 0.8 Ajuste parámetros etapa1: tiempo (VEGA).....	148
Figura 0.9 Ajuste parámetros etapa1: HVperc (VEGA) (a).....	149
Figura 0.10 Ajuste parámetros etapa1: HVperc (VEGA) (b)	150
Figura 0.11 Ajuste parámetros etapa1: gd, ESS y card (VEGA)	150
Figura 0.12 Ajuste parámetros etapa2: HVperc (VEGA) (a).....	151
Figura 0.13 Ajuste parámetros etapa2: HVperc (VEGA) (b)	152
Figura 0.14 Ajuste parámetros etapa2: gd, ESS y card (VEGA)	153
Figura 0.15 Ajuste parámetros etapa1: tiempo (NSGA-II)	155
Figura 0.16 Ajuste parámetros etapa1: HVperc (NSGA-II) (a)	155
Figura 0.17 Ajuste parámetros etapa1: HVperc (NSGA-II) (b)	156
Figura 0.18 Ajuste parámetros etapa1: HVperc (NSGA-II) (c)	158
Figura 0.19 Ajuste parámetros etapa1: gd, ESS y card (NSGA-II).....	158
Figura 0.20 Ajuste parámetros etapa2: HVperc y tiempo (NSGA-II) (a).....	159
Figura 0.21 Ajuste parámetros etapa2: HVperc y tiempo (NSGA-II) (b).....	160
Figura 0.22 Ajuste parámetros etapa2: tiempo (NSGA-II)	161
Figura 0.23 Ajuste parámetros etapa2: HVperc (NSGA-II)	162
Figura 0.24 Ajuste parámetros etapa2: gd, ESS y card (NSGA-II).....	163
Figura 0.25 Ajuste parámetros etapa1: tiempo (MOEA/D) (a)	165
Figura 0.26 Ajuste parámetros etapa1: tiempo (MOEA/D) (b)	166
Figura 0.27 Ajuste parámetros etapa1: HVperc (MOEA/D) (a)	166
Figura 0.28 Ajuste parámetros etapa1: HVperc (MOEA/D) (b)	167
Figura 3.60 Ajuste parámetros etapa1: tiempo (MOEA/D) (c)	167
Figura 0.30 Ajuste parámetros etapa1: HVperc (MOEA/D) (c).....	168
Figura 0.31 Ajuste parámetros etapa1: HVperc (MOEA/D) (d)	169
Figura 0.32 Ajuste parámetros etapa1: gd, ESS y card (MOEA/D).....	170
Figura 0.33 Ajuste parámetros etapa2: tiempo (MOEA/D)	171
Figura 0.34 Ajuste parámetros etapa2: HVperc (MOEA/D) (a)	171
Figura 0.35 Ajuste parámetros etapa2: HVperc (MOEA/D) (b)	172
Figura 0.36 Ajuste parámetros etapa2: HVperc (MOEA/D) (c).....	174

Figura 0.37 Ajuste parámetros tamaño de vecindario: HVperc (MOEA/D).....	175
Figura 0.38 Ajuste parámetros tamaño de vecindario: gd y ESS (MOEA/D).....	176

Tabla 1.1 Instancias a resolver en la investigación (Beasley, 1990).....	8
Tabla 1.2 Técnicas de manejo de restricciones.....	16
Tabla 1.3 Taxonomía de los enfoques multiobjetivo	20
Tabla 3.1 Detalles de Hardware y Software	51
Tabla 3.2 valores de los parámetros en el ejemplo	62
Tabla 3.3 Estadísticos prueba de alcance inst0	65
Tabla 3.4 Métricas, alcance promedio inst2	66
Tabla 3.5 Mejores resultados por método de cruce-mutación (hipervolumen).....	78
Tabla 3.6 Análisis cualitativo: selección de un método de manejo de restricciones.....	88
Tabla 3.7 Valores de los parámetros etapa1.....	92
Tabla 3.8 Valores de los parámetros etapa2 (SPO).....	97
Tabla 3.9 Ajuste parámetros etapa2: selección no determinista de una tasa de mutación.....	99
Tabla 3.10 Ajuste parámetros etapa2: Prueba de rangos de signos de Wilcoxon (SPO).....	100
Tabla 3.11 Valores finales de los parámetros	102
Tabla 4.1 Implementación 1: estadísticos del tiempo (s)	106
Tabla 4.2 Implementación 1: prueba Wilcoxon tiempo.....	107
Tabla 4.3 Ajuste de tiempo para MOEA/D y VEGA	108
Tabla 4.4 Magnitud de la variación al corregir el tiempo	110
Tabla 4.5 Prueba de Wilcoxon Media vs Mediana (HVperc)	115
Tabla 4.6 Orden de desempeño de las técnicas (HVperc).....	116
Tabla 4.7 Prueba Wilcoxon MOEA/D vs NSGA-II (HVperc).....	117
Tabla 4.8 HVperc inst0 mediana.....	119
Tabla 4.9 Métricas inst1 mediana	120
Tabla 4.10 Métricas inst2 mediana	121
Tabla 4.11 Métricas inst3 mediana	122
Tabla 4.12 Métricas inst4 mediana	123
Tabla 4.13 Resumen estadístico resultados finales	124
Tabla 4.14 Rankin con respecto a la mediana de cada métrica (mayor a menor desempeño).....	125
Tabla 4.15 Estado del arte, fuente: (Jin, Qu, & Atkin, 2016).....	127
Tabla 4.16 Comparativo con el estado del arte	129
Tabla 0.1 Ejemplo: Prueba de rangos de signos de Wilcoxon.....	139
Tabla 0.2 Ejemplo: Prueba de Kruskal-Wallis para k muestras independientes.....	142
Tabla 0.3 Valores de los parámetros etapa2 (VEGA)	150
Tabla 0.4 Ajuste parámetros etapa2: Prueba de Wilcoxon (VEGA) (a).....	151
Tabla 0.5 Ajuste parámetros etapa2: Prueba de Wilcoxon (VEGA) (b).....	153
Tabla 0.6 Ajuste parámetros etapa1: Prueba de rangos de signos de Wilcoxon (NSGA-II).....	157

Tabla 0.7 Valores de los parámetros etapa2 (NSGA-II).....	159
Tabla 0.8 Ajuste parámetros etapa2: tiempo (NSGA-II comparado con SPO y VEGA)	161
Tabla 0.9 Ajuste parámetros etapa2: Prueba de Wilcoxon HVperc (NSGA-II)	162
Tabla 0.10 Ajuste parámetros etapa2: Prueba de Wilcoxon, gd (NSGA-II).....	163
Tabla 0.11 Valores de los parámetros etapa1 (MOEA/D).....	164
Tabla 0.12 Ajuste parámetros etapa1: prueba de Kruskal-Wallis (MOEA/D)	168
Tabla 0.13 Valores de los parámetros etapa2 (MOEA/D).....	170
Tabla 0.14 Ajuste parámetros etapa2: Prueba de Wilcoxon (MOEA/D) (a).....	173
Tabla 0.15 Ajuste parámetros etapa2: prueba de Kruskal-Wallis (MOEA/D)	173
Tabla 0.16 Ajuste parámetros etapa2: Prueba de Wilcoxon (MOEA/D) (b)	174
Tabla 0.17 Ajuste parámetros tamaño de vecindario: Prueba de Wilcoxon (MOEA/D).....	176
Tabla 0.18 Estadísticos descriptivos de implementación final: HVperc	176
Tabla 0.19 Estadísticos descriptivos de implementación final: gd	177
Tabla 0.20 Estadísticos descriptivos de implementación final: ESS	178
Tabla 0.21 Estadísticos descriptivos de implementación final: card	178

Introducción

Al adentrarse al estudio de los problemas de optimización es imposible no percatarse de la existencia de éstos en prácticamente toda disciplina científica. Sobre la ciencia de los bienes, los recursos y la asignación, dice Lomelí & Rumbos (2001):

“La enseñanza de métodos cuantitativos para economistas ha sufrido grandes cambios durante la última década. En el pasado era posible, para un estudiante de licenciatura en economía, tener una idea clara de su profesión con un mínimo de conocimientos cuantitativos. [...] Lo importante era el saber utilizarla para los fines requeridos sin profundizar mucho en su origen.

La labor del economista es, y debe seguir siendo, la del estudio y creación del conocimiento económico; sin embargo, dado que el lenguaje utilizado para la descripción de este conocimiento es el de las matemáticas, no queda más remedio que conocerlo a fondo para evitar su uso incorrecto. Es en este sentido que han cambiado los cursos de matemáticas para economistas.”

Para conocer el reto que representa una disciplina como la optimización basta con solo reconocer la gran diversidad de los problemas formulados, involucrando diferentes tipos de funciones objetivo y restricciones, resultando en ocasiones sumamente difíciles de resolver. Además, una gran mayoría de

aplicaciones del mundo real involucra más de un objetivo o criterio de desempeño; por lo tanto el concepto de optimización multiobjetivo pasó a ser un área de gran interés. Aunado al hecho de que tiene un impacto sobre la esfera del conocimiento económico-financiero en la actualidad, siendo así el punto de partida: los fenómenos financieros.

El problema del portafolio de inversión consiste en la selección de un conjunto de activos de inversión. Los objetivos en general tienen que ver con la diversificación de la inversión: la minimización del riesgo y la maximización del retorno.

Pero, ¿cuál es la trascendencia de un fenómeno económico como la actividad financiera en la disciplina de la optimización? (Lomelí & Rumbos, 2001) Una gran cantidad de problemas a resolver por el matemático tienen su origen en algún fenómeno económico, los casos más claros siendo la teoría de juegos y las matemáticas financieras. Así que, el primer paso para resolver un problema será la identificación de las particularidades y a continuación, poner en uso las herramientas de investigación disponibles ante el desarrollo del objetivo general: la identificación de una metaheurística eficiente para resolver un problema multiobjetivo (MOP por sus siglas en inglés) financiero en particular, el del portafolio de inversión.

Este problema fue estudiado y formulado desde 1952 por Markowitz (1952) quien propuso por primera vez un modelo matemático para este tipo de problemas, se trata de definir un portafolio eficiente, es decir, aquel que tiene un

mínimo riesgo para un retorno dado o, equivalentemente un portafolio con un máximo retorno para un nivel de riesgo dado. Sin duda es una abstracción muy simple pero a la vez útil para empezar a entender las variables de interés y, de hecho, es posible utilizar técnicas de programación cuadrática para encontrar soluciones óptimas. Pero el problema se vuelve complejo cuando se integran ciertas prohibiciones, o reglas del mercado real, sobre los objetivos: el problema de optimización multiobjetivo restringido conlleva al uso de técnicas metaheurísticas. Este aspecto constituye el punto de partida y los objetivos del presente proyecto: implementar y estudiar el desempeño de diferentes técnicas metaheurísticas de optimización multiobjetivo, particularmente Algoritmos Evolutivos Multiobjetivo, para la solución del problema de Portafolio de Inversión.

El presente documento se organiza como se describe a continuación: el capítulo uno proporciona el marco teórico, relacionando el problema de inversión con el área de optimización multiobjetivo. En un segundo capítulo, se incluye el desarrollo de los métodos para la selección de poblaciones no dominadas, así como la descripción de las características y el modo operativo de las metaheurísticas manejadas. El tercer capítulo presenta la implementación del ajuste de parámetros, se presentan las métricas que se adoptaron. Posterior a ello, la etapa experimental es detallada y la configuración de los parámetros que se determinó. Finalmente, en el cuarto capítulo se presenta la parte experimental del proyecto, es decir, las corridas finales de los algoritmos con los parámetros ajustados, así como su análisis estadístico

comparativo. Se incluye también una sección de comparación de los resultados con el estado del arte.

Los apéndices están conformados por detalles de la metodología estadística usada, análisis a fondo acerca de algunos conceptos empleados en la investigación, así como información más detallada acerca de la metodología de ajuste de parámetros.

1 Marco teórico

En el campo de las finanzas, la inversión consiste en el posicionamiento de un cierto capital en el mercado, de tal forma que en un cierto periodo de tiempo se logre obtener una recompensa o ganancia. A menudo la proporción que representa ésta última con respecto al capital inicial se le llama tasa de interés o de ganancia. También se relaciona el retorno esperado de una inversión con un cierto riesgo de perder parte del capital inicial, riesgo que es típicamente mayor a medida que la promesa de ganancia es mayor. Partiendo de la concepción anterior de la inversión, se puede aseverar que una cartera o portafolio de inversión hace referencia a un conjunto de activos financieros que se cotizan en el mercado bursátil, en los cuales un agente económico decide invertir su dinero. Para una inversión como la del portafolio se tendrá como objetivos obtener la mayor ganancia posible y, por otro lado, que el riesgo de pérdida sea bajo. Esta estrategia de buscar repartir el riesgo al combinar diferentes instrumentos se conoce como diversificación del portafolio. El problema de optimización resultante es, por lo tanto, bi-objetivo.

1.1 El problema del portafolio de inversión

1.1.1 El modelo de Markowitz

En la presente investigación el proceso de selección de activos para un portafolio será trabajado utilizando el modelo matemático propuesto por Markowitz (1952). Éste se plantea originalmente, como un problema de optimización monobjetivo, en el cual se pretende maximizar el retorno esperado sujeto a un determinado riesgo establecido o equivalentemente minimizar el riesgo para un retorno establecido. En el marco de la presente investigación, estos valores establecidos serán denotados como metas y modificándolos, puede ser resuelto como un MOP.

Desde la perspectiva de este proyecto, el modelo de Markowitz (1952) es un MOP, donde el vector objetivo está formado por las funciones de retorno esperado y riesgo (a menudo calculado como la varianza del retorno). De acuerdo a lo anterior, en el problema del portafolio de inversión, $x = [x_1, x_2, \dots, x_N]$ representa un conjunto de inversión específico llamado cartera y x_i es la proporción del capital que se invierte en el activo i , $f_1(x)$ cuantifica el riesgo de la cartera y $f_2(x)$ el retorno esperado. El modelo correspondiente se formula de la siguiente manera:

$$\text{Minimizar } \left\{ f_1(x) = \left[\sum_{i=1}^N \sum_{j=1}^N x_i x_j \sigma_{ij} \right], \quad - f_2(x) = \left[\sum_{i=1}^N x_i \mu_i \right] \right\}$$

sujeto a

$$\sum_{i=1}^N x_i = 1$$

$$\begin{aligned} x_i &\geq 0, & i &= 1, \dots, N. \\ 0 &\leq x_i \leq 1, & i &= 1, 2, \dots, N \end{aligned}$$

Donde x_i es la proporción del capital que se invertirá en el activo i , σ_{ij} es la covarianza entre el activo i y el j , μ_i es el retorno esperado del activo i y N es el número de activos disponibles en el mercado. La restricción de igualdad indica que el total del capital disponible debe ser invertido en la cartera.

1.1.2 Extensiones: otros modelos

En el estado del arte, es posible encontrar diversas variantes y modificaciones del modelo de Markowitz, esto con objeto de considerar otras características presentes en el mundo real. Ponsich, et al. (2013) hacen una clasificación de los tipos de variantes que se han propuesto (Figura 1.1), basado en tres aspectos que son: integración de restricciones, adición de objetivos y el criterio de optimización Sharpe's Ratio.

Figura 1.1 Extensiones del modelo de Markowitz (Ponsich, López Jaimes, & Coello, 2013)

Integración de restricciones del mundo real	Inclusión de otros objetivos	Optimización del Sharpe's Ratio
<ul style="list-style-type: none"> • Restricciones de techo-suelo: asignación de un máximo y mínimo valor de ciertos activos que puede tener la cartera • Peso total asignado a las clases de activos: asignación a un sector específico una proporción fija del capital • Restricciones de cardinalidad: Número de activos que es necesario incluir en la cartera • Restricciones de lote de transacción mínimo: La cantidad invertida en un activo debe ser un múltiplo del lote de transacción mínimo, que representa el volumen más pequeño de este valor que se puede comprar • Restricciones del volumen de transacción: éstas se refieren a la fijación de un umbral de variación en el peso de un activo con respecto a una asignación anterior 	<ul style="list-style-type: none"> • Nuevas formas de representar el riesgo • Uso de distribuciones distintas a la normal multivariante para el retorno esperado • Consideración de preferencias de inversionistas individuales • Consideración de las fluctuaciones de un activo de la cartera en torno a su media, debido a la volatilidad del mercado • Introducción de la semivariancia, la cual representa la probabilidad de posibles pérdidas. • Value-at-Risk (VaR): el α – cuantil (típicamente $\alpha = 5\%$) de la distribución de las pérdidas de una cartera. El VaR representa la pérdida máxima en una cartera que se puede esperar con un nivel de confianza $100(1 - \alpha)$ • Desempeño a 12 meses de un activo • Rendimiento de tres años de un activo • Dividendo anual de una cartera • Clasificación en estrellas de S & P • Costos de transacción 	<ul style="list-style-type: none"> • Es un indicador que mide cuánto exceso de beneficio por unidad de riesgo entrega una determinada cartera y está parametrizado por un valor R_f que indica el retorno observado (o deseado) de una cartera libre de riesgo. • Una consecuencia importante del uso de este criterio de selección es el hecho de que regiones específicas del frente de Pareto resultan ser más relevantes. • La inclusión de este criterio de selección en el proceso de optimización permite, por lo tanto, centrarse en esta región donde el Sharpe's Ratio alcanza su valor máximo.

1.2 Las instancias

Para evaluar la confianza y eficiencia de las técnicas implementadas, existen varios bancos de instancias ya estudiados en la literatura especializada. En este proyecto, se propone trabajar con cinco mercados de capital propuestos en (Beasley, 1990), los cuales fueron elegidos debido a que son ampliamente utilizados en la aplicación práctica dentro de la literatura. La información se muestra en la Tabla 1.1 y fue obtenida a partir de stocks de marzo de 1992 a septiembre de 1997.

Tabla 1.1 Instancias a resolver en la investigación (Beasley, 1990)

ID	País	Instancia	Tamaño (N)
0	Hong Kong	Hang Seng31	31
1	Alemania	DAX85	85
2	UK	FTSE89	89
3	USA	S&P98	98
4	Japón	Nikkei225	225

Para cada instancia se tiene la siguiente información de entrada:

- Retorno promedio semanal de cada activo
- Desviación estándar del retorno semanal
- Matriz de correlación entre activos (P)

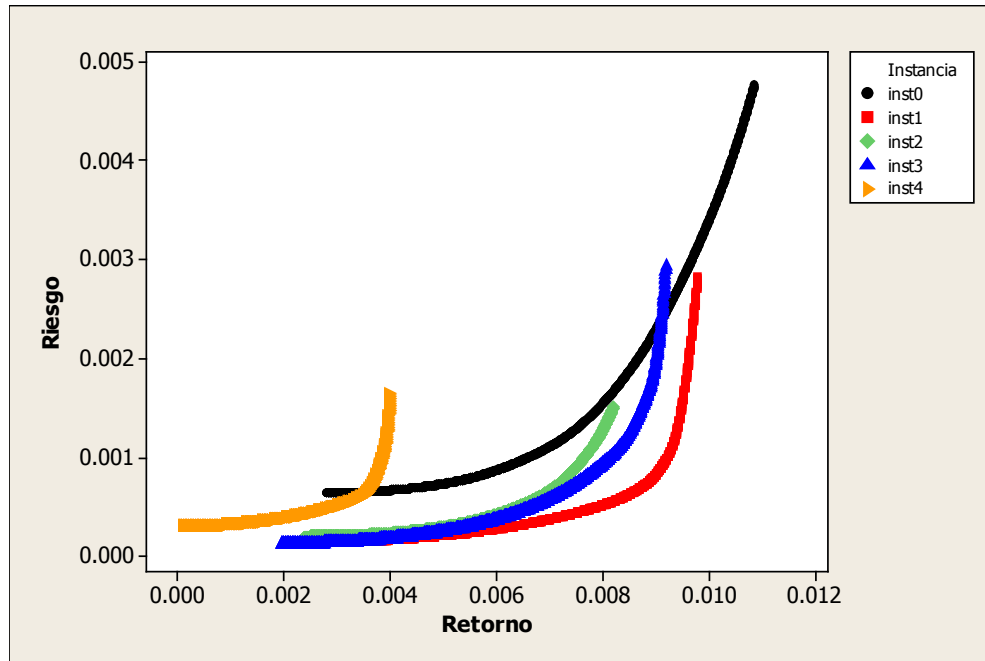
Para evaluar las funciones objetivo solo faltan las covarianzas entre activos, las cuales serán calculadas de la siguiente manera:

$$\sigma_{ij} = \rho_{ij}s_i s_j$$

Donde s_i es la desviación estándar del activo y ρ_{ij} es el índice de correlación entre los activos i y j .

Se dispone de los frentes de Pareto reales de todas las instancias, lo cual es una ventaja para poder evaluar el desempeño de las técnicas de optimización probadas más adelante.

Figura 1.2 Frentes de Pareto reales para las instancias estudiadas



1.3 Optimización: generalidades

Antes de comenzar a profundizar en el entorno multiobjetivo, es de suma trascendencia definir con precisión los conceptos de optimización fundamentales, el presente apartado está enfocado en ello. Los elementos que conforman un problema de este tipo, como lo son los objetivos, los tipos de restricciones, así como las principales herramientas metodológicas que se han desarrollado para resolver tales problemas, son el primer paso para comprender el desarrollo de la optimización y consecutivamente las técnicas metaheurísticas. En particular los algoritmos genéticos juegan el papel primordial en la presente investigación, por tanto es necesario también el desarrollo conceptual de su forma canónica monobjetivo.

1.3.1 Optimización monobjetivo

El problema clásico de optimización continua consiste en: dada una función $f: \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ llamada objetivo, encontrar un elemento $x_0 = [x_1, x_2, \dots, x_n]$

sujeto a

$$g_1(x) \leq 0$$

$$g_2(x) \leq 0$$

\vdots

$$g_k(x) \leq 0$$

$$x \in \Omega$$

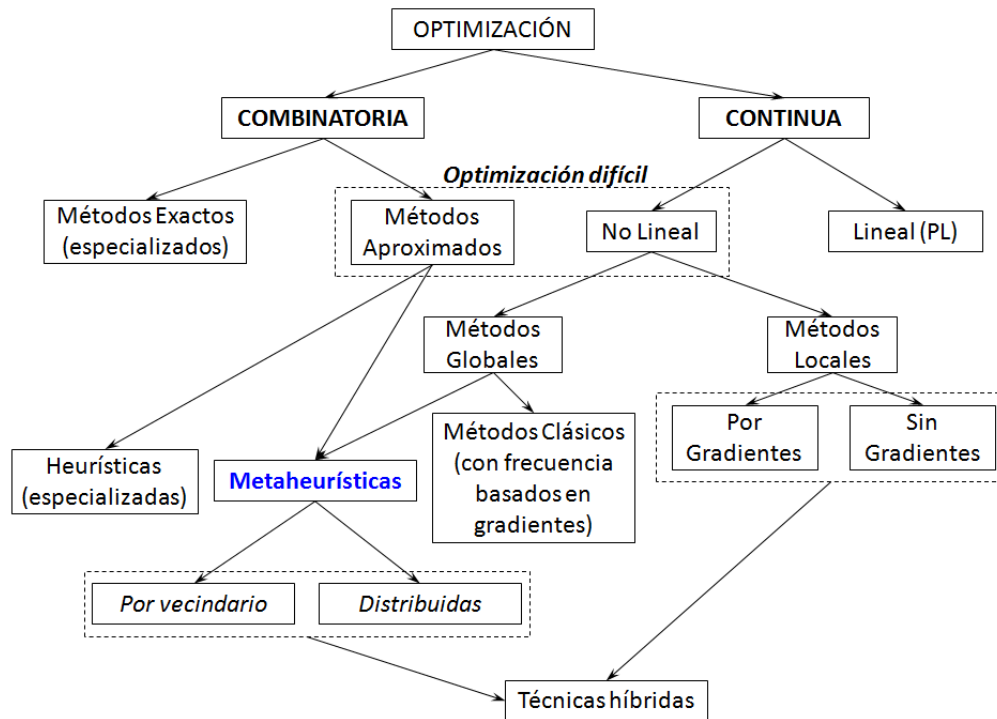
Tal que $f(x_0) \leq f(x)$ para toda $x \in \Omega$ (donde Ω es el espacio de búsqueda) si se está minimizando o $f(x_0) \geq f(x)$ para toda x si se está maximizando. A partir de este punto y sin pérdida de generalidad, por optimizar se estará refiriendo a minimizar, ya que maximizar a f es equivalente a minimizar $-f$.

De acuerdo a la definición anterior, aquellos problemas donde $\Omega \subseteq \mathbb{R}^n$ son llamados de optimización continua, por otro lado, si $\Omega \subseteq \mathbb{Z}^n$ entonces se trata de uno de optimización entera o problema de combinatoria.

En la Figura 1.3 se muestra una clasificación de las técnicas y métodos que se utilizan para abordar los problemas de optimización. La presente investigación se concentrará en métodos para problemas continuos no lineales, los cuales, en el estado del arte, también son referidos como de optimización difícil, debido a que no existen algoritmo eficiente o método que pueda garantizar óptimos

globales de manera exacta para tales problemas. En estos casos, se recurre frecuentemente al uso de técnicas metaheurísticas para optimización global. Una clase de estas metaheurísticas son los Algoritmos Evolutivos.

Figura 1.3 Técnicas de optimización (Collette & Siarry, 2004)



1.3.2 El paradigma de los algoritmos genéticos

Un algoritmo genético (GA por sus siglas en inglés) es un tipo de algoritmo evolutivo, el cual se trata de una metaheurística utilizada inicialmente para la resolución de problemas de optimización monobjetivo, basada en una metáfora de la teoría de la evolución de Darwin (1859). Los GAs fueron propuestos por Holland (1975) y su funcionamiento radica en:

- Hacer análoga la representación de una solución y el valor de la función objetivo (fitness) evaluado en ésta, a la información genética

de un individuo y a su aptitud (adaptación al medio ambiente) respectivamente.

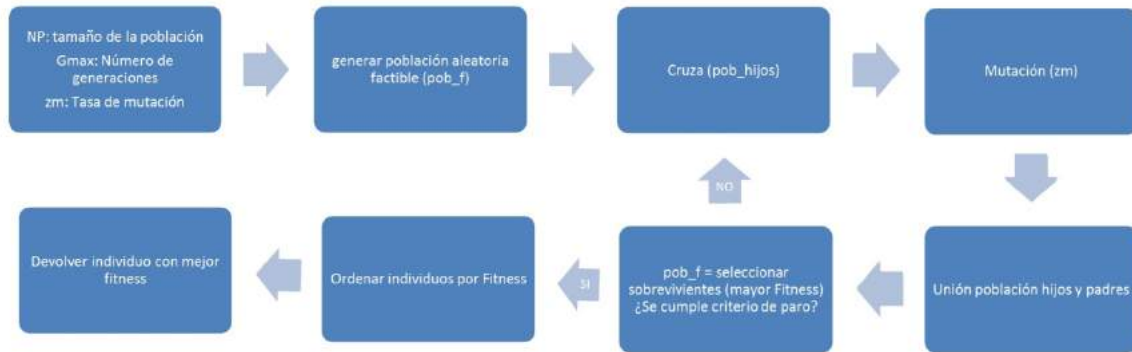
- Mantener un conjunto de individuos (vectores soluciones) llamado población.
- Generar una interacción genética entre individuos (combinación de padres), durante la cual se producen otros más llamados hijos, los cuales tendrán características similares a sus padres pero introducen variaciones en el patrimonio genético.
- Permitir la posibilidad de perturbaciones de la información genética generación tras generación de individuos (tasa de mutación).
- Aplicar un proceso análogo al concepto de selección natural, que promueve la supervivencia de los individuos con mejores cualidades en cada nueva generación.

En consecuencia, si se permite que las soluciones interactúen de esta forma, cuando se decida detener el proceso utilizando algún criterio específico, se obtendrá un conjunto de soluciones de buena calidad. Además, debido a la etapa de mutación, coexistirá una diversidad de individuos en la población, evitando que ésta última se reduzca a un solo tipo de soluciones, es decir, que el proceso de búsqueda se quede atrapado en un óptimo local del problema. La Figura 1.4 muestra el proceso mediante el cual opera un GA.

Respecto a la calidad de las soluciones finales y por tanto la cercanía al óptimo global de éstas, bajo ciertas condiciones específicas, existe una

probabilidad positiva que la combinación de bloques constructores permita llegar al óptimo global (Goldberg, 1989).

Figura 1.4 Algoritmo Genético monobjetivo



Algoritmo 1.1 Genético monobjetivo (minimizar)

Variables de entrada

NP: tamaño de la población

Gmax: Número de generaciones

zm: Tasa de mutación

generar población aleatoria factible pob_f

para generación en {0, ..., Gmax-1}:

 para i en pob_f:

 selección de padres

 cruza

 si |pob_hijos| < |pob_f|:

 agregar hijo a pob_hijos

 mutación a una tasa especificada zm

 p = unión población hijos y padres

 p_nueva = selección(p)

 pob_f = p_nueva

devolver mejor individuo de pob_f

El Algoritmo 1.1 es el que se ha utilizado en la presente investigación¹. Se debe recordar que para que un algoritmo genético sea considerado como tal, tendrá que incluir las etapas de selección, recombinación, mutación y elitismo.

1.3.3 Manejo de restricciones

Los Algoritmos Genéticos (GAs por sus siglas en inglés) en su versión canónica son técnicas de búsqueda sin restricciones. De ahí surge el interés por implementar mecanismos que les permitan trabajar con restricciones tanto de igualdad como de desigualdad así como lineales y no lineales (Mezura Montes, 2001).

Muchas estrategias de manejo de restricciones han sido desarrolladas, como se describen en (Coello, 2002) y (Michalewicz & Attia, 1994). La Tabla 1.2 da una visión general sobre ellas.

¹ Se utiliza la notación NP únicamente en los pseudocódigos y Ψ en el resto del documento para hacer referencia al tamaño de la población.

Tabla 1.2 Técnicas de manejo de restricciones

Manejo de restricciones	
Funciones de penalización <ul style="list-style-type: none"> • Pena de muerte • Penalizaciones estáticas • Penalizaciones dinámicas • Penalizaciones adaptativas • Recocido simulado • Penalizaciones co-evolutivas • Algoritmo genético segregado • Penalización basada en factibilidad 	Otras técnicas <ul style="list-style-type: none"> • Representaciones y operadores especiales • Algoritmos de reparación • Separación de restricciones y objetivos • Métodos híbridos

La mayor parte de estas estrategias requieren incorporar la información de infactibilidad de un individuo (normalmente se considera la distancia a la zona factible o el número de restricciones violadas) a su aptitud correspondiente, para penalizar y que en el proceso de selección perseveren tanto las soluciones factibles como aquellas prometedoras (cercanas a ser factibles). De las técnicas restantes, la más popular consiste en establecer una estricta diferencia entre soluciones factibles e infactibles.

1.4 Optimización multiobjetivo

En esta sección se pretende hacer un análisis de las particularidades que implica un problema de carácter multiobjetivo, la forma en la que se abordan estos problemas, los métodos para su resolución, así como la manera de cuantificar la calidad de las soluciones encontradas.

1.4.1 Generalidades

Ya se ha mencionado en la sección 1.3.1 el modelo general de optimización monobjetivo, de igual modo es posible definir uno de tipo multiobjetivo, a diferencia de que en tal caso la función objetivo tendrá la forma $\vec{f}: \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^m$, es decir, asociamos a cada $x = [x_1, x_2, \dots, x_n]$ un vector de funciones reales $\vec{f}(x) = [f_1(x), f_2(x), \dots, f_m(x)]$ y la meta es optimizar tales funciones reales simultáneamente. Por tanto el problema consiste en:

$$\text{minimizar } \vec{f}: \Omega \rightarrow \mathbb{R}^m$$

sujeto a

$$g_1(x) \leq 0$$

$$g_2(x) \leq 0$$

$$\vdots$$

$$g_k(x) \leq 0$$

$$x \in \Omega$$

Es decir, encontrar un elemento $x_0 = [x_1, x_2, \dots, x_n]$ tal que $\vec{f}(x_0) \preceq \vec{f}(x)$ para toda $x \in \Omega$ (donde $\Omega \subseteq \mathbb{R}^n$ es el espacio de búsqueda), donde $a \preceq b$ significa que “a es mejor que b” bajo algún criterio. Pero dado que f es una función vectorial, es necesario definir él o los criterios bajo los cuales $f(x_0) \preceq f(x)$, y a partir de ello si además x_0 es única.

La dificultad inherente a los problemas de optimización multiobjetivo es que, en general, los objetivos $f_1(x), f_2(x), \dots, f_m(x)$ son antagónicos, es decir que es

imposible optimizar simultáneamente todas las f_i al mismo tiempo ya que la mejora de una de ellas puede implicar el deterioro de otras más. En este contexto, es preciso redefinir el concepto de optimalidad desde otra perspectiva. La noción más aceptada de “óptimo” en el entorno de MOPs es la propuesta originalmente por Francis Ysidro Edgeworth en 1881 y generalizada posteriormente por Vilfredo Pareto en 1896. Se busca no una, sino un conjunto P de soluciones de la forma $\mathbf{y} = [y_1, y_2, \dots, y_n]$, llamadas soluciones no dominadas, que se definen de la siguiente manera:

Dado un MOP de la forma antes presentada, sea $\Omega \subset \mathbb{R}^n$ el espacio factible, decimos que una solución $\mathbf{y} \in \Omega$ donde $\mathbf{y} = [y_1, y_2, \dots, y_n]$ es no dominada si $\nexists \mathbf{x} \in \Omega, f_i(\mathbf{x}) \leq f_i(\mathbf{y}) \ \forall i \in \{1, 2, \dots, m\}$ y al menos para alguna $j \ f_j(\mathbf{x}) < f_j(\mathbf{y})$.

Algunos autores llaman tales soluciones óptimo de Edgeworth-Pareto, pero es más común denominarla simplemente óptimo de Pareto, o soluciones Pareto-óptimas. El conjunto P es conocido como el conjunto de Pareto y es el que se suele presentar como la solución al MOP. Su imagen en el espacio de los objetivos es el frente de Pareto real (PF o PFtrue por sus siglas en inglés) como se muestra en el ejemplo de la Figura 1.5².

² se usaron 100 puntos factibles aleatorios

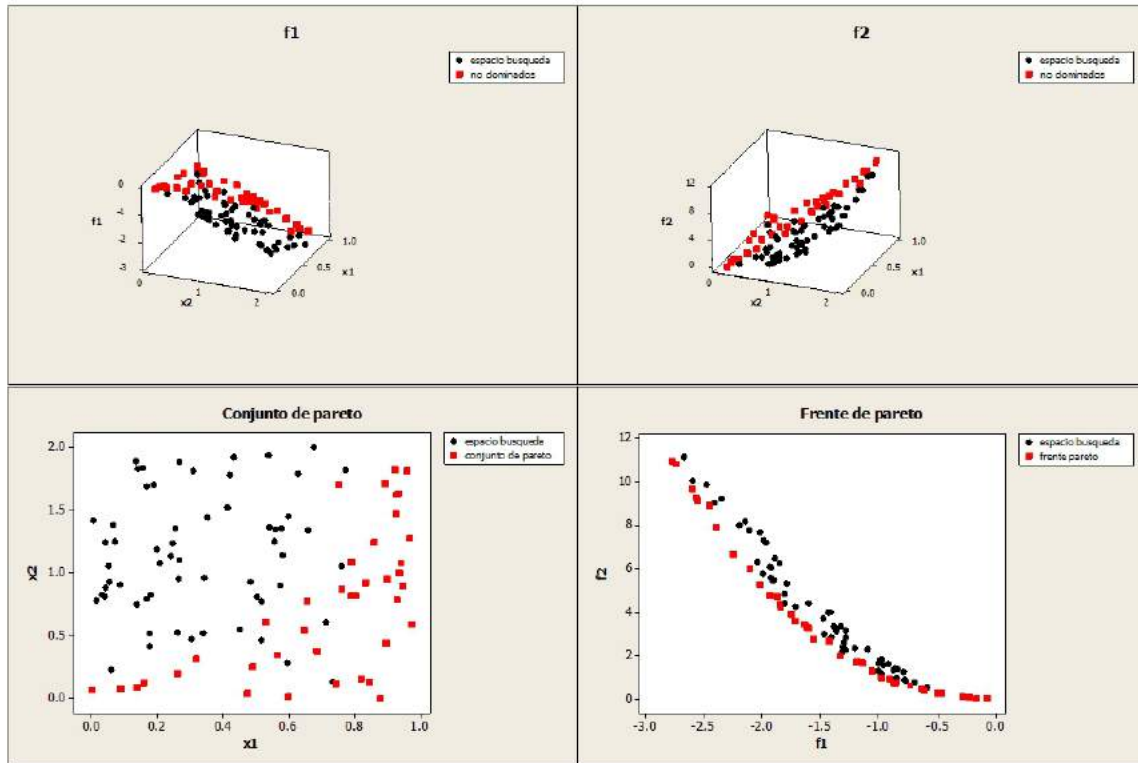
Figura 1.5 Diferencia entre conjunto y frente de Pareto

$$\min f(x) = [f_1(x_1, x_2) = -x_1 - x_2, f_2(x_1, x_2) = (x_1 + 2)^2 + (x_2)^2]$$

s. a.

$$0 \leq x_1 \leq 1$$

$$0 \leq x_2 \leq 2$$



1.4.2 Taxonomía de los enfoques multiobjetivo

Tabla 1.3 Taxonomía de los enfoques multiobjetivo

Técnicas multiobjetivo	No basadas en Pareto (Técnicas de IO, multi→mono)	Ordenamiento lexicográfico
		Suma lineal de pesos
		Vector Evaluated Genetic Algorithm (VEGA)
		ϵ -restricciones
		Satisfacción de metas
		Teoría de juegos
	Basadas en Pareto (aptitud de una solución: criterio de dominancia)	MOEAs 1era generación (NSGA, MOGA, y NPGA)→Dominancia
		MOEAs 2da generación (NSGA-II, NPGA-II, SPEA-II, PAES)→Dominancia, elitismo
		Enfoques no poblacionales (AMOS)
	Basados en otros paradigmas (Muchos objetivos)	Basados en métricas (SMS-EMOA) (2005)
		Descomposición (MOEA/D) (2007)

En la Tabla 1.3 se muestran los dos principales enfoques estratégicos de las técnicas multiobjetivo:

- I. Las técnicas no basadas en Pareto típicamente reducen el MOP en un problema monobjetivo, donde la aptitud se define de acuerdo a criterios como: importancia de los objetivos, metas a alcanzar para los mismos, etc.

- II. Aquellas basadas en Pareto miden la aptitud de una solución usando un criterio de dominancia³.

1.4.3 Métricas

Si el trabajo de medir la calidad de los resultados obtenidos por un algoritmo en el contexto monobjetivo es sencillo (distancia mínima o promedio al óptimo, desviación estándar de las soluciones, etc.), lo mismo no es cierto en un marco multiobjetivo. Efectivamente, la meta es doble: obtener una buena convergencia hacia el frente de Pareto real y, al mismo tiempo, tener una buena dispersión de las soluciones no dominadas producidas. Por lo tanto, se han propuesto diferentes métricas que tratan de medir la calidad del frente de Pareto aproximado (PF_{aprox} por sus siglas en inglés) por algún algoritmo en términos de esos dos criterios además de esfuerzo computacional, tales como:

- Número de llamadas a la función objetivo o tiempo de cómputo,
- Número de puntos en el frente aproximado,
- Relacionado con la convergencia al frente real (PF_{True}): tasa de error (Van-Veldhuizen, 1999), Distancia Generacional (gd), Distancia Generacional Inversa (igd), hipervolumen (Coello, Van-Veldhuizen, & Lamont, 2002).
- Relacionado con la dispersión de las soluciones en el espacio de los objetivos: *Efficient Set Spacing* (ESS) (Schott, 1995), hipervolumen.

³ Una desventaja crucial de estas técnicas es que, el criterio de dominancia de Pareto pasa a ser irrelevante a medida que el número de objetivos aumenta: el número de soluciones no dominadas tiende a crecer junto con los objetivos, lo cual dificulta discriminar soluciones bajo el criterio de dominancia. para mayores detalles consultar: (Coello, Van-Veldhuizen, & Lamont, 2002).

En el marco del presente proyecto, se seleccionarán y definirán específicamente algunas de estas métricas para interpretar la calidad de las diferentes metaheurísticas probadas (véase el capítulo 1).

2 Técnicas Metaheurísticas en Optimización Multiobjetivo

En este capítulo se describirán las técnicas utilizadas en el presente proyecto para resolver el problema del portafolio de inversión. Cabe mencionar que, antes de presentar las heurísticas de optimización multiobjetivo, se empieza definiendo los algoritmos de selección de soluciones no dominadas, necesarios para posteriormente implementar dichas técnicas.

2.1 Algoritmos de selección de Poblaciones No Dominadas (PNDs)

En el entorno multiobjetivo, será necesario utilizar un método eficiente para seleccionar soluciones no dominadas de un conjunto de soluciones factibles, lo cual es un proceso que puede resultar muy costoso computacionalmente hablando. Por tal motivo se pusieron a prueba diferentes algoritmos que realizan dicho proceso de clasificación para elegir el más eficiente en las heurísticas estudiadas.

Para la determinación de dominancia entre dos soluciones expresadas en el espacio de los objetivos, es decir, únicamente la determinación de la relación de dominancia entre dos soluciones, se han utilizado dos algoritmos.

El primero, se trata de un comparador exhaustivo, su modo operativo consiste en revisar siempre la relación tricotómica entre todos los elementos de las dos soluciones uno a uno, a continuación declarar verdadera o falsa la dominancia de la primera solución sobre la segunda.

El segundo algoritmo que se ha implementado, es el propuesto por Bentley, Kennet, & David (1993). Es más eficaz, debido a que no se comparan todos los valores de la solución, sino que el proceso se detiene cuando se tiene suficiente información para concluir respecto al criterio de dominancia.

Algoritmo 2.1 Comparador de soluciones Bentley (Bentley, Kennet, & David , 1993)

```

Variables
m = número de objetivos
sol1, sol2 en el espacio de los objetivos
i = 0
incomp = Falso
mientras (i <= m-1 y sol1[i] == sol2[i]):
    i = i + 1
si i > m-1:
    resultado = 3
de otro modo:
    si sol1[i] < sol2[i]:
        para j en [i+1, ..., m-1]:
            si sol2[j] < sol1[j]:
                incomp = Verdadero
                rompe ciclo
        si incomp == Verdadero:
            resultado = 2
        de otro modo:
            resultado = 0
    de otro modo:
        para j en [i+1, ..., m-1]:
            si sol1[j] < sol2[j]:
                incomp = Verdadero
                rompe ciclo
        si incomp == Verdadero:
            resultado = 2
        de otro modo:
            resultado = 1
devolver resultado

para la variable resultado:
# 0 => sol1 domina a sol2
# 1 => sol2 domina a sol1
# 2 => sol1 y sol2 son no dominadas entre sí
# 3 => sol1 y sol2 son iguales

```

Nótese que en una sola implementación del Algoritmo 2.1 se puede obtener una de las cuatro posibilidades de relación entre dos soluciones a y b :

- Que la solución a domine a b ,
- Que la solución b domine a a ,
- Que a y b sean no dominadas entre sí,
- Que a y b sean iguales.

A continuación se detallan los tres algoritmos de selección probados, los cuales tienen como función, a partir de un conjunto de soluciones dadas (en el espacio de los objetivos) seleccionan únicamente las no dominadas.

2.1.1 El algoritmo exhaustivo (A1)

Este algoritmo A1 consiste tan solo en tomar cada solución i y buscar si alguna de todas las demás soluciones la dominan, si ninguna lo hace, se agrega la i -ésima a la Población No Dominada (PND). Se implementó utilizando, de igual forma, el comparador exhaustivo, lo cual lo hace un algoritmo con un costo computacional muy alto, $O(\Psi^2 m)$ donde Ψ es el tamaño de la población a clasificar.

2.1.2 Algoritmo versión II (A2)

Esta versión, es mostrada en el Algoritmo 2.2. A primera vista se puede notar que evita hacer muchas comparaciones, ya que permite obtener conclusiones con menos información. Agrega soluciones a la PND y eventualmente las elimina si posteriormente otra solución más las domina, sin la necesidad de

compararla con toda la población. En el transcurso de la ejecución, la PND mejora hasta finalizar quedándose solo con soluciones no dominadas. Este algoritmo se implementó con el comparador exhaustivo y aunque su complejidad en el peor de los casos es igual que la del anterior $O(\Psi^2 m)$, es mejor en el caso promedio.

Algoritmo 2.2 Selector de conjunto de Pareto A2

```

Variables
conjunto de soluciones a clasificar pt
tamaño de la población inicial total lp
población no dominada pnd
i = 1
j = 0
paro = 0
mientras paro==0:
    si pnd[j] domina a pt[i]:
        si i == (lp-1):
            paro = 1
        de otro modo:
            i = i + 1
            j = 0
    de otro modo:
        si pt[i] domina a pnd[j]:
            elimina pnd[j]
            si j >= (|pnd| -1):
                agregar a pnd el elemento pt[i]
                si i == (lp-1):
                    paro = 1
                de otro modo:
                    i = i + 1
                    j = 0
            de otro modo:
                si j >= (|pnd| -1):
                    agregar a pnd el elemento pt[i]
                    si i == (lp-1):
                        paro = 1
                    de otro modo:
                        i = i + 1
                        j = 0
                de otro modo:
                    j = j + 1
devolver pnd

```

2.1.3 Clasificador de Bentley (M3)

Se trata del selector de soluciones no dominadas propuesto por Bentley, Kennet, & David (1993), de acuerdo a los autores, quienes lo mencionan como “Un Algoritmo Heurístico Efectivo”. Efectivamente, el costo total esperado para una población de tamaño Ψ es $O(\Psi \log^{m-1} \Psi)$.⁴

Algoritmo 2.3 Selector de conjunto de Pareto M3 (Bentley, Kennet, & David , 1993)

```

Variables
conjunto de soluciones a clasificar pt
tamaño de la población inicial total lp
TopMax = 0
pnd = {pt[TopMax]}
para i en [1, ..., lp-1]:
    j = 0
    mientras j <= TopMax:
        si pnd[j] domina a pt[i]:
            agrega al principio de la lista a pnd[j]
            j = TopMax + 2
        de otro modo si pt[i] domina a pnd[j]:
            borrar pnd[j]
            TopMax = TopMax - 1
        de otro modo si pnd[j] pt[i] no son dominados entre sí:
            j = TopMax + 2
        de otro modo:
            j = j + 1
    si j == (TopMax + 1):
        TopMax = TopMax + 1
        agrega pt[i] al final de pnd
devolver pnd

```

En el Algoritmo 2.3, para realizar la comparación entre dos soluciones, se utiliza el Algoritmo 2.1 propuesto por los mismos autores, dado que entrega la solución de la forma requerida.

⁴ Para mayores detalles consultar (Bentley, Kennet, & David , 1993)

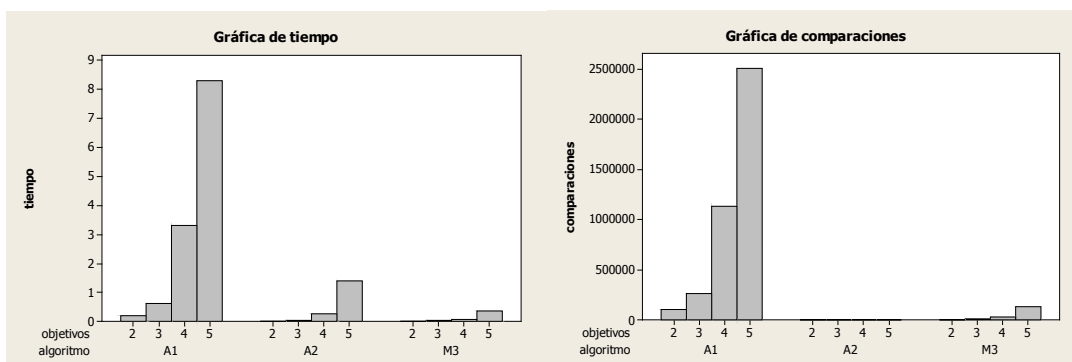
2.1.4 Análisis de eficiencia

Se realizaron experimentos computacionales preliminares, para comprobar la eficiencia de los tres algoritmos descritos anteriormente. Las características del hardware y software para la implementación de la sección actual son las siguientes:

- Lenguaje: Python 2.7.8 MSC v.1500 32 bit (Intel)] on windows7
- Procesador: AMD A4 5300 3.4 GHz
- Memoria: RAM 8 GB

La primera prueba realizada para comparar el funcionamiento de los algoritmos, consiste en determinar los elementos no dominados de una población de 4,000 soluciones, generadas aleatoriamente en el espacio $[0,1]^m$ donde m es el número de objetivos, variando entre 2 y 5, los resultados obtenidos se muestran de forma gráfica en la Figura 2.1. De principio, es posible notar cómo, al incrementar el número de objetivos, aumenta el tiempo de ejecución muy rápido para A1, para A2 esto es más suave y M3 en esta prueba tendrá el desempeño mejor (menor tiempo y tendencia).

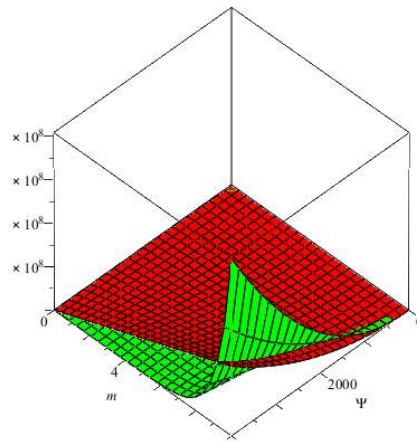
Figura 2.1 Tiempo (s) de ejecución, Llamadas al comparador de soluciones, selectores de PNDs



Además, en el número de veces en que el algoritmo llama a su respectivo comparador de soluciones, se puede distinguir la superioridad del comparador en M3 ya que a pesar de ser llamado más veces que el correspondiente en A2, M3 es mejor en tiempo⁵.

Recordando las complejidades computacionales respectivas de los tres algoritmos probados, graficadas en la Figura 2.2, se puede apreciar que para las dimensiones (no mayor a cinco) y el tamaño de población (4,000) utilizado está por debajo la función logarítmica. Pero si, para el mismo rango en Ψ se manejaran 9 objetivos o más, la cuadrática está ahora por debajo; de acuerdo a lo anterior se debe tener cuidado en qué algoritmo preferir en función de la necesidad de cuántos objetivos y tamaños de población se pretende manejar.

Figura 2.2 Gráficas Rojo: $\Psi^2 m$ Verde: $\Psi[\log_2(\Psi)]^{m-1}$



⁵ M3 no sería mejor con el comparador exhaustivo, éste último habría que llamarlo un mucho mayor número de veces, ya que M3 requiere conocer los 4 estados posibles de dominancia

2.2 Algoritmo Genético basado en una suma ponderada de los objetivos

En esta sección se abordará una de las primeras propuestas para afrontar un problema de optimización multiobjetivo. Se trata de una manera sencilla de plantear el problema desde la perspectiva monobjetivo e implementar una técnica clásica para resolverlo (en este caso un GA). Esta técnica constituye un buen punto de partida y referencia para estudiar los alcances de otros métodos, médula de la presente investigación, los cuales fueron creados específicamente para atacar los problemas de interés en este proyecto.

2.2.1 La noción de función de agregación

Una función de agregación, suma lineal de pesos, combinación convexa o suma ponderada de objetivos (SPO), está definida por Zhang & Li (2007) de la siguiente manera:

Dado un vector de pesos $\lambda = (\lambda_1, \dots, \lambda_m)^T$, con $\lambda_i \geq 0$ para toda $i = 1, \dots, m$ y $\sum_{i=1}^m \lambda_i = 1$, se puede reformular el problema de optimización multiobjetivo como un problema de optimización escalar:

$$\begin{aligned} \text{minimizar } g^{ws}(x|\lambda) &= \sum_{i=1}^m \lambda_i f_i(x) \\ \text{sujeto a } x &\in \Omega \end{aligned}$$

Donde Ω es el espacio factible, y x es el vector de variables. De esta forma, se está reduciendo el problema original a un problema de un solo objetivo. Se obtiene un conjunto de diferentes vectores óptimos de Pareto al resolver el

problema escalar varias veces, utilizando múltiples vectores de pesos. Esta técnica sencilla tiene, sin embargo, desventajas importantes, entre las cuales las más relevantes son:

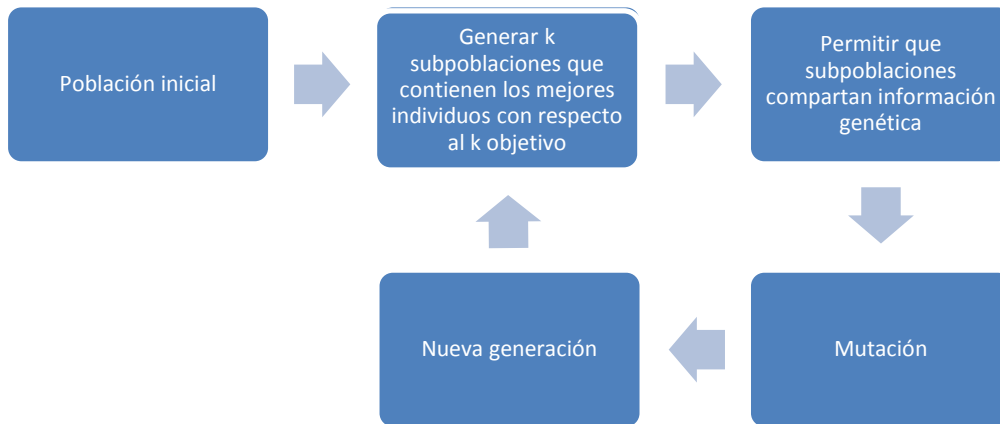
- Para cada solución no dominada deseada, es necesario resolver un problema de optimización escalar, por lo que la aproximación del frente de Pareto aproximado requiere de múltiples ejecuciones del método de solución.
- Debido a la reformulación del problema, no se pueden alcanzar regiones no convexas del frente de Pareto.
- La selección adecuada de los vectores de pesos a utilizar es problemática, ya que, según la naturaleza (lineal o no) del problema, una distribución uniforme de dichos vectores no garantiza que las soluciones obtenidas se distribuyan uniformemente a lo largo del frente

2.3 La estrategia Vector Evaluated Genetic Algorithm (VEGA)

Una de las primeras propuestas hechas específicamente para MOPs es la estrategia Vector Evaluated Genetic Algorithm (VEGA), la cual fue presentada por Schaffer (1985). Su funcionamiento está basado en el paradigma de los algoritmos genéticos como su nombre lo indica, pero la característica principal yace en la parte más polémica en el entorno multiobjetivo: el modo en el que se le asigna aptitud a un individuo. Como se puede observar en la Figura 2.3 se realiza una fase de clasificación en la que se crean k subpoblaciones con los mejores individuos con respecto a cada uno de los objetivos. Por lo tanto, los

individuos muy aptos con respecto a más de un objetivo podrían aparecer en más de una subpoblación.

Figura 2.3 Etapas de un algoritmo basado en la estrategia VEGA



El Algoritmo 2.4 es la versión implementada en la presente investigación, es acertado destacar los siguientes dos puntos:

- I. Se tiene como meta en un MOP determinar las soluciones no dominadas, en cada generación éstas tendrán la característica de que, por ser buenas en todos los objetivos, aparezcan en múltiples subpoblaciones y por ende tener mayor posibilidad de ser seleccionadas para cruza.
- II. La etapa de creación de subpoblaciones permite especializar el proceso de optimización para objetivos aislados, a continuación cuando las soluciones son combinadas, dado que hay un proceso de selección por medio de torneo binario, dentro de esta sub-etapa (ver sección 1.3.2) el

elitismo por dominancia estará presente, lo cual le permitirá al algoritmo ser más dinámico⁶.

Algoritmo 2.4 VEGA Canónico

```

variables de entrada

NP: Tamaño de la población
Gmax: Número de generaciones
obj: Número de objetivos

inicio
#tamaño de subpoblación
tam_sub = NP/obj
generar población aleatoria factible pob_f
para generación en [0, ..., Gmax-1]:
    m_pool = ∅
    para k en [0, ..., obj-1]:
        ordenar pob_f de menor a mayor con respecto al valor del objetivo k
        sub_pob = ∅
        i = 0
        mientras tamaño de sub_pob < tam_sub:
            agregar pob_f[i] a sub_pob
            i = i + 1
        m_pool = m_pool U sub_pob
    ordenar aleatoriamente a m_pool
    mientras cardinalidad(pob_hijos) < NP:
        seleccionar padres de m_pool
        cruza(padre1, padre2) #se producen 2 hijos
        agregar hijo a pob_hijos
    mutacion de pob_hijos
    pob_f = pob_hijos
devolver no dominados de pob_f

```

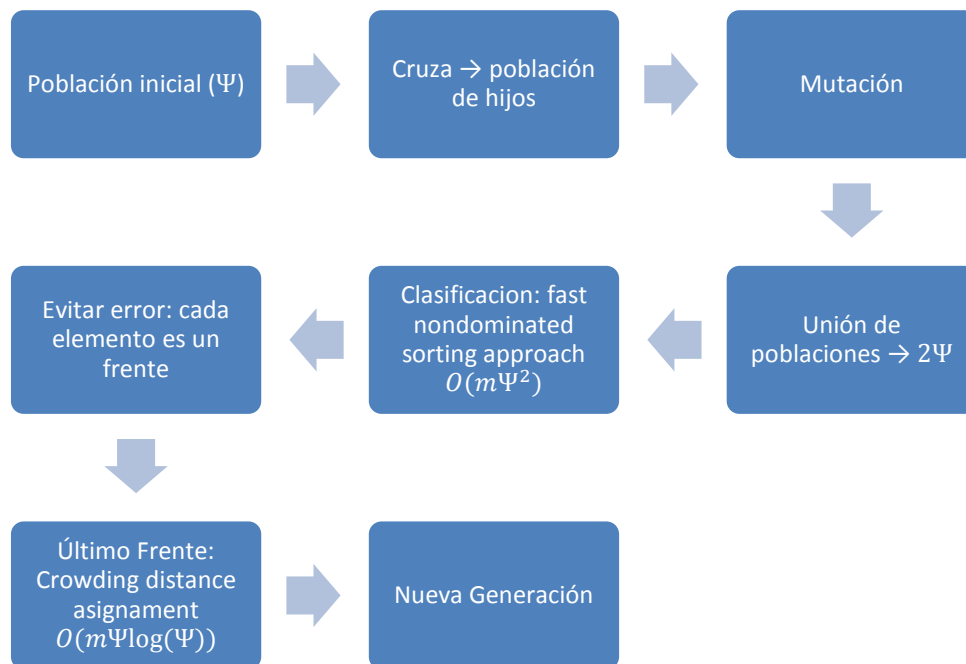
⁶ Es preciso aclarar que la técnica en sí, no muestra como tal una etapa donde se discriminen soluciones por criterio de dominancia, sin embargo al utilizar el método de selección por medio de torneo binario para la presente investigación el efecto Pareto está presente de forma indirecta.

2.4 La estrategia Nondominated Sorting Genetic Algorithm II (NSGA-II)

El método Elitist Nondominated Sorting Genetic Algorithm (NSGA-II) fue propuesto por Deb, Pratap, Agarwal, & Meyarivan (2002) y tiene una complejidad de $O(m\Psi^2)$. Se trata de una versión más eficiente y elitista del NSGA anteriormente propuesto por Srinivas & Deb (1995).

El NSGA-II es una estrategia en la cual es crucial el concepto de dominancia de Pareto en la evolución de las poblaciones en el tiempo. De hecho, la clasificación de soluciones no dominadas implica un costo computacional relativamente muy alto. A grandes rasgos, las etapas que conforman esta estrategia se muestran en la Figura 2.4.

Figura 2.4 Etapas de un algoritmo basado en la estrategia NSGA-II



2.4.1 Jerarquía de dominancia

Es de gran trascendencia el mecanismo de selección en el NSGA-II a partir de haber realizado las etapas de cruce y mutación, es nombrada Jerarquía de Dominancia y la describe el siguiente procedimiento:

1. Se forma la unión entre población de padres e hijos, la cual tendrá un tamaño 2Ψ , se detonará como población global (PG).
2. Los elementos no dominados del total de PG conforman la subpoblación (o frente) de rango 1.
3. Se remueven de PG los elementos anteriormente mencionados, ahora se seleccionan de la población restante los individuos no dominados (soluciones que sólo serán dominadas por las subpoblaciones de rango menor), este subconjunto ahora tiene soluciones un rango por encima del subconjunto del anterior paso.
4. Se repite el paso 3 hasta que los elementos que queden en PG sean no dominados entre sí, éstos últimos conforman la última subpoblación (la cual es la de mayor rango en la clasificación). Se puede observar un ejemplo de cómo quedaría una clasificación en la Figura 2.5.

Una vez concluido el anterior proceso de jerarquización se continúa con una etapa de selección de la siguiente manera:

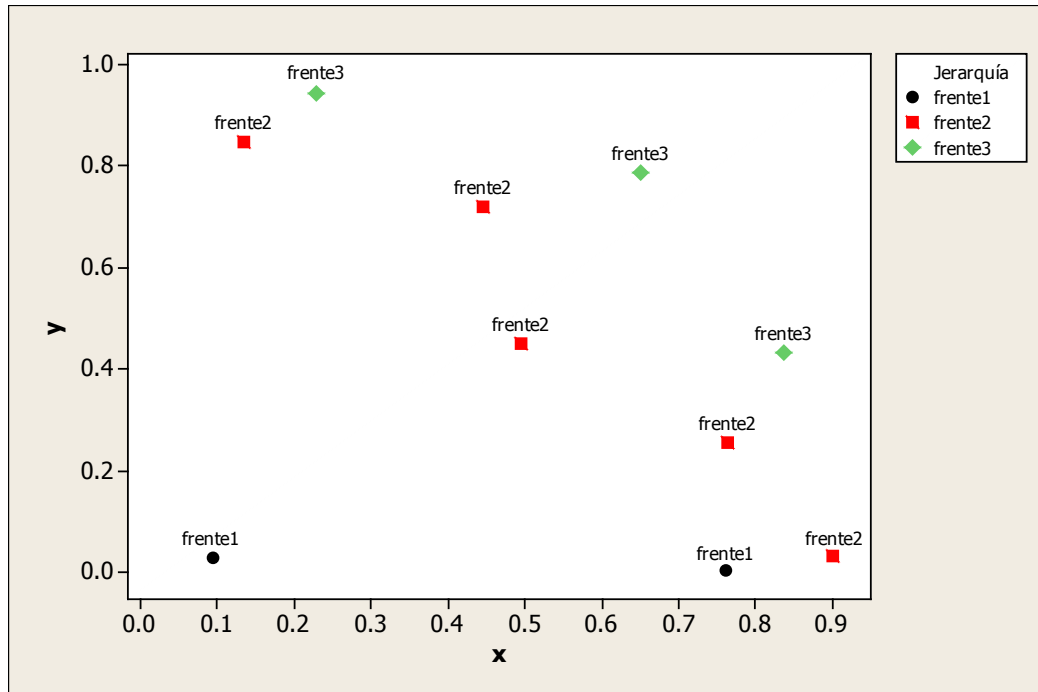
1. Teniendo la anterior partición de PG , se forma un nuevo conjunto que constituirá la nueva generación (NG) y que además debe tener N

elementos, parece pertinente seleccionar las soluciones calificadas como de menor rango (1), así éstas se incorporan a (NG).

2. Si NG aún no alcanza la cardinalidad apropiada, se incorporan los frentes siguientes de rango superior mientras $|NG| \leq \Psi$.
3. En el caso más favorable, la última subpoblación añadida a NG tiene el número de elementos exactos para completar los Ψ elementos requeridos, concluyendo así la etapa de selección, pero eso no sucede muy a menudo.
4. De lo contrario, si añadir una subpoblación implica sobrepasar el tamaño de NG , entonces solo se añaden los elementos necesarios para cumplir la meta de Ψ elementos, pero permanece el problema de cómo seleccionarlos ya que son individuos con iguales aptitudes en términos de dominancia.
5. Sea SG el último subconjunto mencionado en el paso anterior, esta vez la aptitud de los individuos de este conjunto será asignada acorde a una métrica de dispersión llamada Crowding distance, que geométricamente, mide el grado de proximidad entre soluciones vecinas⁷.
6. Se añaden una a una las soluciones de SG a NG de mayor a menor valor para Crowding distance hasta que $|NG| = \Psi$ (sección 2.4.2).
7. Se generan hijos mediante una nueva etapa de cruce y mutación y se regresa al proceso de jerarquización.

⁷ Para mayores detalles consultar (Deb, Pratap, Agarwal, & Meyarivan, 2002)

Figura 2.5 Ejemplo jerarquías de dominancia (minimización)



Algoritmo 2.5 NSGA-II

Gmax: Número de generaciones

NP: Tamaño de la población

inicio

generar pob_f

para generacion en [0, ..., Gmax-1]:

 mientras tamaño de pob_hijos < tamaño de pob_f:

 selección de padres por torneo binario

 cruza(padre1, padre2) #se producen 2 hijos

 agregar hijos a pob_hijos

 mutar pob_hijos

 p_total = pob_f U pob_hijos

 partición(p_total) = ordenar por jerarquía de dominancia (menor a mayor rango)

 p_nueva = \emptyset

 para cada frente en la partición:

 unir frente con p_nueva si cardinalidad(p_nueva) \leq NP:

 elegir individuos de último frente en función de crowding-distance

 pob_f = p_nueva

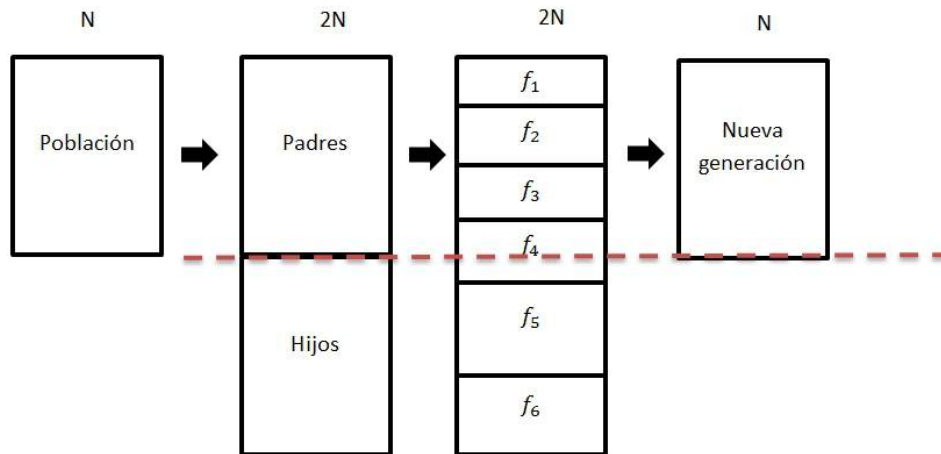
pob_f = no dominados de pob_f

devolver pob_f

2.4.2 Crowding distance assignment

Esta métrica se usa en la estrategia NSGA-II para decidir, en el caso de que sea necesario, qué individuos del último frente capaz de completar una nueva generación deben ser seleccionados. La Crowding Distance (CD) es una aptitud asignada únicamente para los elementos del último frente que no fue posible añadir a la nueva generación, de acuerdo a la Figura 2.6, f_4 sería tal frente.

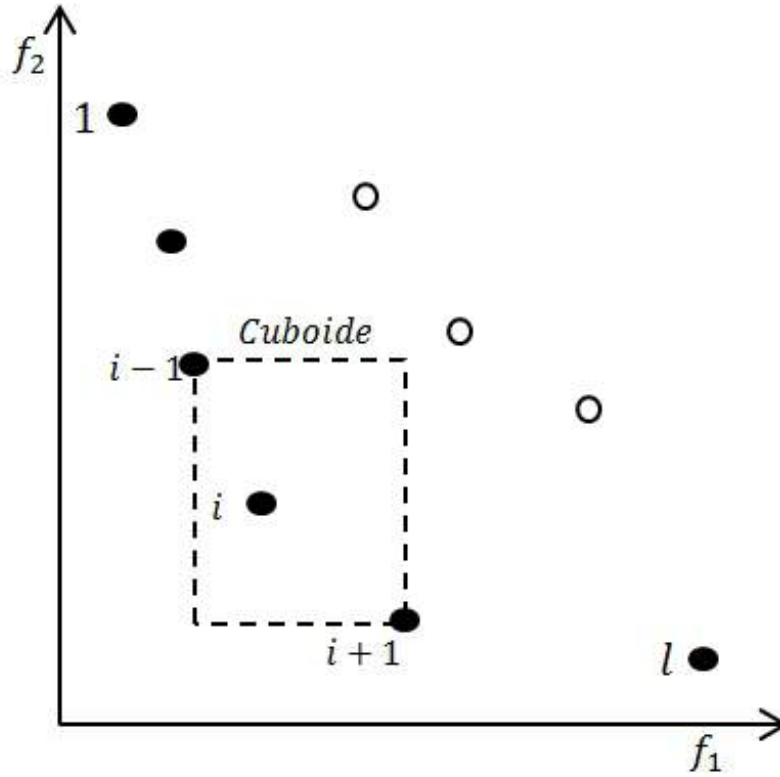
Figura 2.6 Generación de frentes NSGA-II



Así pues, para el frente f_4 se calcula la métrica utilizando el Algoritmo 2.6. La CD es proporcional al perímetro del cuboide cuyos vértices están formados por las soluciones vecinas del individuo considerado, en su mismo frente. La Figura 2.7 ilustra esta definición en un espacio de dos objetivos. A medida que el cuboide ocupe un mayor volumen, se le carga un valor más grande a la CD del i –ésimo individuo, promoviendo su selección ya que está en una región menos densamente poblada. Nótese que, previo al cálculo de CD, es necesario

normalizar los objetivos de tal manera que diferencias de magnitud entre sus valores no afecten la validez del procedimiento.

Figura 2.7 Cálculo del Crowding-distance en un espacio de dos objetivos



Efectivamente, dado que en la etapa del Algoritmo 2.6 donde se aumenta $I[i + 1][m] - I[i - 1][m]$ al valor de CD, lo que se está haciendo es lo siguiente:

$$\Delta cd = \frac{\text{lado } m \text{ del cuboide}}{\text{intervalo que abarcan las soluciones en el objetivo } m}$$

Así este cargo representa más bien una proporción (porcentaje) y no una cantidad expresada en la unidad de medida (magnitud) del objetivo m .

Algoritmo 2.6 Cálculo de Crowding distance

```

obj: número de objetivos
I: frente en el espacio de los objetivos
L: cardinalidad(I)

crd = [0, 0, ..., 0] #crowding distances
para m en [0, 1, ..., obj-1]:
    I = ordenar I con respecto al objetivo m de menor a mayor
    crd[0] = ∞
    crd[L-1] = ∞
    para i en [1, 2, ..., L-2]:
        si I[L-1][m] - I[0][m] ≠ 0:
            I[i][obj] = I[i][obj] + (I[i+1][m] - I[i-1][m]) / (I[L-1][m] - I[0][m])
crd = ordenar crd de menor a mayor
devolver crd

```

2.5 Multiobjective Evolutionary Algorithm based on Descomposition (MOEA/D)

MOEA/D es una técnica propuesta por Zhang & Li (2007), basada en una reformulación del MOP en un conjunto de problemas escalares y que se resuelven simultáneamente, de esta forma el proceso de optimización se opera en un entorno monobjetivo. La estrategia explícitamente descompone el MOP en Q sub-problemas de optimización escalar y se resuelve estos sub-problemas simultáneamente por medio de la evolución de una población de soluciones. Cada sub-problema es parametrizado por un vector de pesos (similar al usado en la función de agregación presentada en la sección 2.2.1), que permite orientar la búsqueda hacia cierta región del frente de Pareto. En cada generación, la población se compone de la mejor solución encontrada hasta ese momento para cada sub-problema. El MOEA/D proporciona una manera simple

pero eficiente de introducir la aproximación por descomposición en el cálculo evolutivo multiobjetivo.

Resolver problemas de optimización escalar en lugar de directamente el problema de optimización multiobjetivo (MOP por sus siglas en inglés) en su conjunto, puede implicar que, cuestiones tales como la asignación de la aptitud de las soluciones (también conocida como fitness) y el mantenimiento de la diversidad –que causan dificultades para otros MOEAs no basados en descomposición– podrían llegar a ser más fáciles de manejar en el marco de la técnica de este apartado (Zhang & Li, 2007).

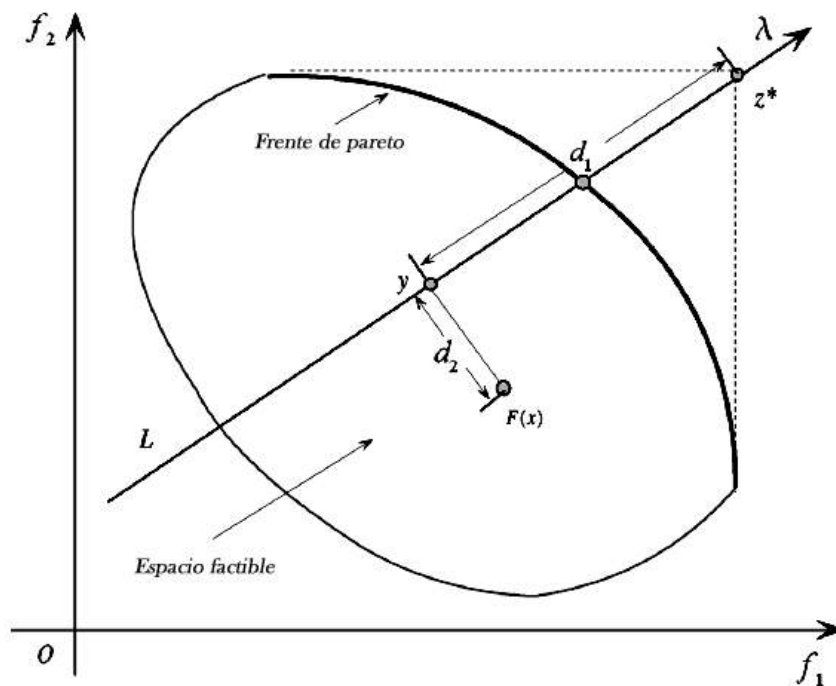
MOEA/D tiene una menor complejidad computacional en cada generación que otras técnicas de optimización multiobjetivo, dado que no implica ningún proceso de clasificación de Pareto: usando una pequeña población es capaz de producir pocas soluciones finales muy uniformemente distribuidas. El ajuste de los vectores de pesos representa sin embargo una tarea difícil, particularmente para problemas de 3 o más objetivos.

2.5.1 Boundary Intersection (BI) Approach

Existen distintas propuestas para efectuar la descomposición (o escalarización) necesaria para la implementación del MOEA/D. En la presente investigación se ha experimentado particularmente con una de las tres consideradas en (Zhang & Li, 2007), conocida como Boundary Intersection (BI) Approach.

Bajo ciertas condiciones de regularidad, el PF de un MOP (de maximización) continuo es parte de la frontera más alta a la derecha de un objetivo fijado a alcanzar (Figura 2.8). Geométricamente, estos enfoques BI tienen como objetivo, como su nombre lo indica, encontrar puntos de intersección entre el límite superior y un conjunto de líneas que emanan del punto ideal (denotado como z^*). Si estas líneas están distribuidas de manera adecuada, se puede esperar que los puntos de intersección resultantes proporcionen una buena aproximación, uniforme, de la totalidad del PF. Cabe mencionar que este enfoque es capaz de aproximar FP no cóncavos.

Figura 2.8 Boundary Intersection (BI) Approach, fuente: (Zhang & Li, 2007)



Matemáticamente, se tienen los siguientes sub-problemas de optimización escalar:

$$\begin{aligned}
 &\text{minimizar } g^{bip}(\mathbf{x}|\lambda, z^*) = d_1 + \theta d_2 \\
 &s. a. \\
 &\mathbf{x} \in \Omega \\
 &\text{donde} \\
 &d_1 = \frac{\|(z^* - F(\mathbf{x}))^T \lambda\|}{\|\lambda\|} \\
 &d_2 = \|F(\mathbf{x}) - (z^* - d_1 \lambda)\| \\
 &\theta > 0 \text{ parámetro de penalización}
 \end{aligned}$$

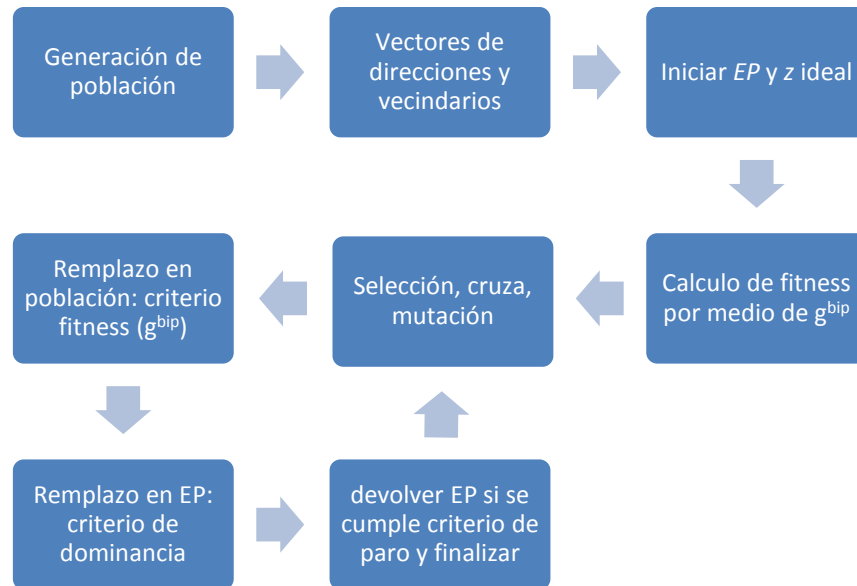
Donde el parámetro θ define el desvío (d_2) con respecto a la dirección λ (λ es determinada por el vector de pesos parametrizando el sub-problema considerado) en la búsqueda del problema escalar actual, contra la velocidad a la que la solución se irá acercando al PF (d_1).

Las relaciones de vecindad entre los sub-problemas se definen en función de las distancias entre vectores de pesos. Así pues, las soluciones óptimas a dos sub-problemas vecinos deben ser similares. Cada sub-problema (es decir, la función BI particular a un vector λ) se optimiza en MOEA/D mediante el uso de la información sólo de sus sub-problemas vecinos.

Es importante mencionar que la estrategia MOEA/D requiere además de un espacio de memoria o archivo de almacenamiento llamado *EP*, en el cuál se administran soluciones no dominadas (ver Algoritmo 2.7). Se ha propuesto

delimitar el tamaño del archivo por medio de Crowding distance⁸ (ver sección 2.4.2).

Figura 2.9 Etapas de un algoritmo basado en la estrategia MOEA/D



Las principales fases de un algoritmo basado en la estrategia MOEA/D se muestran en la Figura 2.9, y el pseudocódigo que incluye las características propias de la implementación en la presente investigación es el Algoritmo 2.7.

Algoritmo 2.7 MOEA/D

```

NP ← tamaño de la población
Gmax ← número de generaciones (criterio de paro)
T ← número de vecinos para cada individuo (preferiblemente par)

inicio
  generar vectores de peso # e.g. lam = {[0, 1], [0.1, 0.9], [0.2, 0.8], ... , [1, 0]}
  generar población pob_f
  agregar no dominados de pob_f a EP
  z = los mejores valores para cada objetivo en EP
  cálculo de aptitud de individuos en pob_f (g_bip)
  para generación en Gmax hacer:
  
```

⁸ El procedimiento de delimitación se explica a detalle en el apéndice: El problema del tamaño de archivo EP

```
para cada individuo en pob_f hacer:
    seleccionar dos vecinos
    cruzar produciendo dos hijos
    mutación de los dos hijos
    evaluación función objetivo de los dos hijos
    actualiza z ideal (si algún hijo es mejor en algún objetivo)
    #reemplazo de vecinos e individuo si hijo es mejor
    Para cada hijo hacer:
        Para cada vecino de individuo hacer:
            calcular el fitness del hijo con el vector de vecino
            Si fitness de hijo es mejor que el de vecino:
                reemplazar al vecino con hijo
        calcular el fitness de hijo con el vector de individuo
        Si el fitness calculado de hijo es mejor que el de individuo:
            reemplazar a individuo con hijo
    si ningún elemento de EP domina a hi:
        se integra hijo a EP
        se eliminan los dominados por hi en EP

devolver EP
```

3 Ajuste de parámetros y manejo de restricciones

Es sabido que uno de los más grandes inconvenientes de los métodos heurísticos es la asignación de los valores adecuados a sus diversos parámetros de operación, este capítulo será primordialmente dedicado al ajuste de dichos parámetros, en los diferentes algoritmos implementados en el marco de este proyecto.

Antes que nada, para evaluar la calidad de un juego de parámetros, se requiere saber medir la calidad de las soluciones encontradas por un algoritmo en el entorno multiobjetivo. Ya se ha discutido anteriormente que la respuesta a un problema de esta naturaleza no consiste en una solución en particular, sino más bien en un conjunto de soluciones eficientes. Asimismo, en el apartado 1.4.3 se mencionaron las características que debe tener tal conjunto para considerarse bueno. Las métricas de desempeño deben evaluar estas características, así que no solo se medirá la cercanía a la solución óptima como en el entorno monobjetivo, esta vez se tratará de un dilema multiobjetivo en sí mismo.

3.1 Métricas utilizadas

Particularmente, en la presente investigación se decidió considerar las siguientes métricas de desempeño, debido a que al tomarlas en cuenta se engloban las características necesarias para considerar una aproximación multiobjetivo de buena calidad.

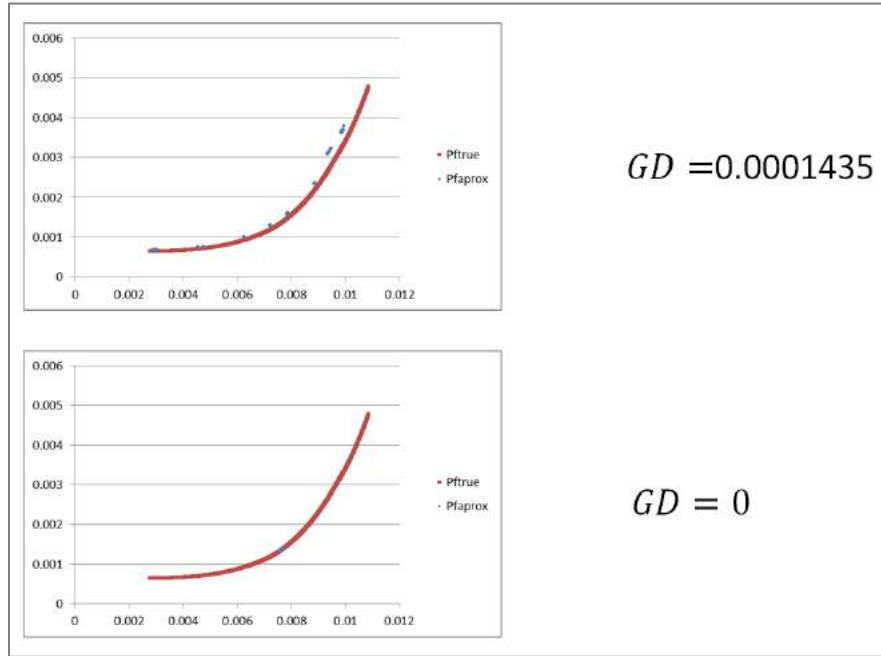
3.1.1 Distancia generacional

La distancia generacional (gd por sus siglas en inglés) se utiliza para saber en promedio qué tan lejos está el frente aproximado (PFaprox) del real (PFtrue), por lo cual se trata de una medida de convergencia. Para el cálculo de esta métrica se necesita el frente real, la fórmula es la siguiente:

$$gd = \frac{\sum_{i=1}^{|PF_{aprox}|} d_i}{|PF_{aprox}|}$$

Donde d_i es la distancia euclidiana (en el espacio de los objetivos) entre la i -ésima solución del PFaprox y la solución más cercana del PFtrue. Un valor cercano a cero indica buena convergencia. La pregunta que surge es ¿por qué no es suficiente tal métrica para explicar la calidad de un frente aproximado? Al observar la Figura 3.1, se muestran dos frentes aproximados, el primero es una buena aproximación y de hecho el valor de gd es bueno (muy cercano a cero), por otro lado, el segundo frente tiene un mejor valor para la métrica, y sin embargo no es una mejor aproximación ya que contiene pocas soluciones muy cercanas, incluso parte del frente real, las cuales no podrían proporcionar indicios de la tendencia de la curva real por completo.

Figura 3.1 Cálculo de la distancia generacional



3.1.2 Efficient Set Spacing

Efficient Set Spacing o simplemente Spacing, describe numéricamente la propagación o dispersión de los vectores de $P_{fitprox}$, la fórmula para su cálculo se muestra a continuación:

$$ESS = \sqrt{\frac{1}{|P_{fitprox}|} \sum_{i=1}^{|P_{fitprox}|} (d_i - \bar{d})^2}$$

$$\text{donde} \begin{cases} d_i = \min_{j \in P_{fitprox}, i \neq j} \left\{ \sum_{p=1}^k |f_p^i - f_p^j| \right\} \\ \bar{d} = E(d_i) \end{cases}$$

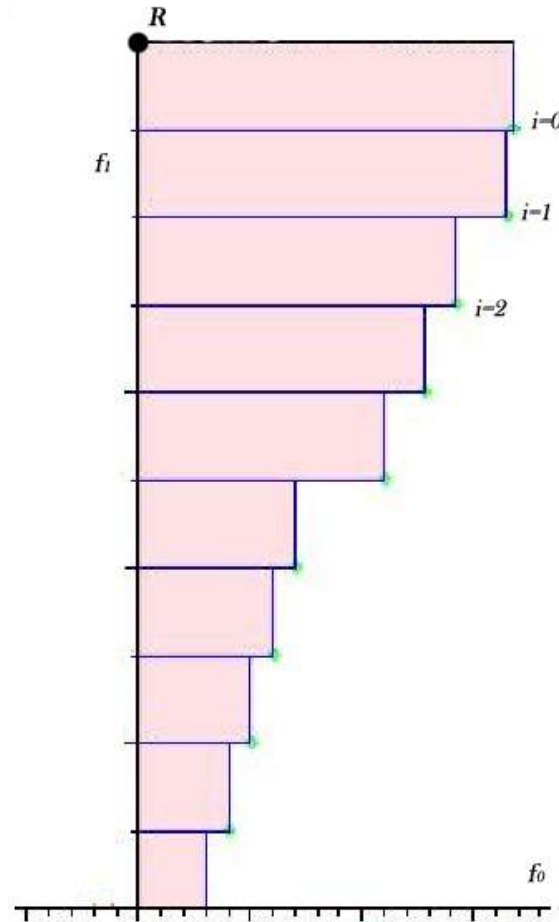
Lo ideal, al igual que gd es un valor cercano a 0.

3.1.3 Hipervolumen

Es definido como el área dominada por PF_{aprox} , con respecto al espacio de los objetivos. Para dos objetivos, esto equivale (Figura 3.2) a la sumatoria de todas las áreas rectangulares delimitadas por algún punto de referencia $R = (f_0(x), f_1(x))$, su cálculo se realiza con:

$$HV = \left\{ \bigcup_i vol_i | vec_i \in PF_{aprox} \right\}$$

Se intenta lograr que este valor sea el mayor posible para concluir que la aproximación es buena. La ventaja del uso del hipervolumen es que es una métrica compatible con la definición de optimalidad de Pareto (Zitzler et al., 2003), es decir que buscar todas las soluciones Pareto óptimas de un MOP es equivalente a maximizar el hipervolumen del frente aproximado correspondiente.

Figura 3.2 Área del hipervolumen en \mathbb{R}^2 

En muchas aplicaciones (incluyendo la presente investigación) la magnitud en la que se mide la función objetivo suele variar en intervalos de reales numéricamente muy pequeños o muy grandes, lo cual dificulta el análisis comparativo y la observación de comportamientos para el hipervolumen. Para un mejor uso de tal métrica se definió una derivada de ésta a la cual se ha nombrado hipervolumen-percent y consiste en el porcentaje o proporción del hipervolumen del frente real que ocupa el hipervolumen aproximado usando el

mismo punto de referencia para ambos frentes⁹, es decir $HV_{perc} = \frac{HV_{aprox}}{HV_{real}} \in [0,1]$, el cual se busca sea cercano a uno¹⁰.

Hasta este punto se han considerado dos métricas especializadas una en la convergencia, otra en la dispersión de forma uniforme, mientras que la tercera puede dar información sobre convergencia y dispersión. En este punto únicamente resta agregar que, para múltiples análisis, será importante conocer el número de puntos aproximados o cardinalidad del frente aproximado.

3.2 Detalles de codificación, hardware y software

Antes de comenzar con el análisis de ajuste cabe señalar que se utilizó codificación real para la representación de las variables. Algunos aspectos importantes en el uso del equipo de cómputo son las características de hardware y software mostrados en la Tabla 3.1.

Tabla 3.1 Detalles de Hardware y Software

Característica	Descripción
Sistema operativo	Ubuntu 16.04.1 LTS 64 bits
Lenguaje de programación	Python 2.7.12
Memoria RAM	16Gb
CPU	Intel Core i7-4770 3.40GHz

⁹Debido a que en la presente investigación $F(x) = (f_0(x), f_1(x)) = (retorno\ esperado, riesgo)$ tiene un frente de Pareto totalmente convexo, el punto de referencia son los valores extremos del mismo, es decir $R = (\max\{retorno\ esperado\}, \min\{riesgo\})$

¹⁰ En algunas ocasiones será expresado en porcentaje.

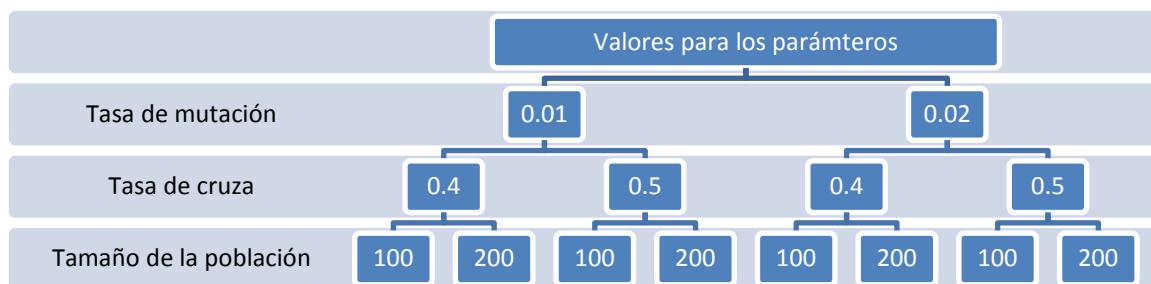
3.3 Metodología para la configuración de parámetros

Se decidió utilizar la configuración de parámetros conocida como diseño factorial completo, el cual consiste en seleccionar un conjunto de valores potenciales para cada parámetro y probar todas las combinaciones de éstos (véase el ejemplo de la Figura 3.3). A veces este método es también llamado “fuerza bruta”, aludiendo a que es la forma exhaustiva o más costosa de hacer una calibración, esto último debido al gran número de experimentos que deben realizarse:

$$s = q_1 q_2 \dots q_p$$

Donde s es el número de experimentos y q_i es el número de valores que deseamos probar para el i – ésimo parámetro. Por otro lado, la información que se obtiene a partir de éste tipo de análisis es más detallada, debido a que se realiza un mayor número de pruebas, lo cual proporcionar mejores indicios acerca del juego de parámetros adecuado.

Figura 3.3 Ejemplo de Análisis factorial completo: dos valores para tres parámetros



Para realizar un análisis exhaustivo, es necesario hacer una delimitación del intervalo en el que varía cada parámetro. Dado que se debe seleccionar un número muy limitado de valores dentro de tales intervalos será necesario elegirlos de manera apropiada y, posteriormente profundizar la búsqueda en el intervalo para encontrar el valor más adecuado. Debido a esto, en la presente investigación se realizaron las etapas de calibración en dos fases llamadas: valores extremos y diseño factorial completo.

3.3.1 Fase 1: valores extremos

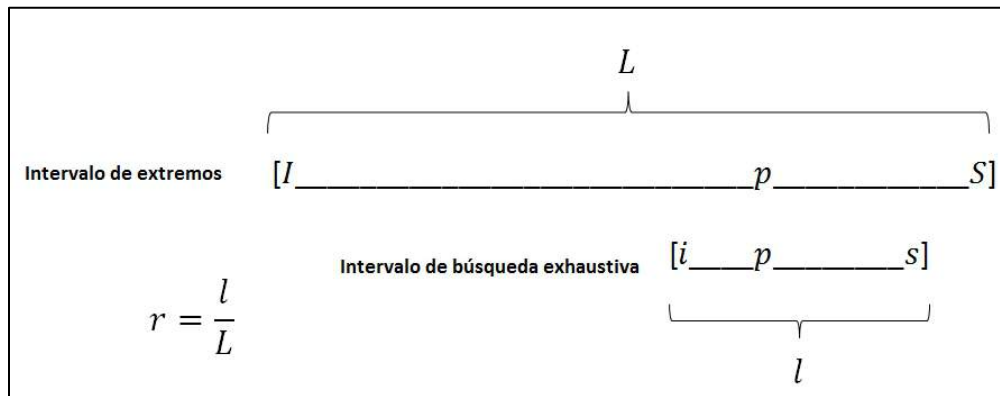
En esta fase se proponen valores fronterizos e intermedios, de este modo se intenta identificar a qué valores es probable se aproximen las mejores opciones para el parámetro. Cada parámetro tiene su propio rango típico de variación, e.g. en la literatura (Coello, 2008) de acuerdo a múltiples experimentos, para una tasa de mutación se sugieren valores no mayores a 0.2, se podría proponer buscar en el intervalo $[0.0001, 0.2]$; mientras que el tamaño de población necesariamente debe ser mayor o igual a 2, por otro lado, para que exista al menos una generación después de la inicial, el tamaño de la población nunca deberá ser mayor a la mitad del número de llamadas a la función objetivo. Si las llamadas a la función objetivo fueran 10,000, se podría proponer para el tamaño de población el intervalo $[6, 2500]$ para permitir que por lo menos sea posible hacer 15 cruces diferentes en cada generación¹¹ (valor inferior) y por otro lado que al menos existan 4 generaciones (valor superior).

¹¹ Esto dado que $C(n, k) = C_k^n$ es el número de subconjuntos de k elementos escogidos de un conjunto con n elementos y $C_2^6 = 15$

3.3.2 Fase 2: diseño factorial completo

Una vez encontrado, para cada parámetro, el mejor valor posible en la fase anterior, denotado como p , lo siguiente consiste en crear un intervalo alrededor de p y realizar la búsqueda amplia final (plan factorial) en este nuevo intervalo, para determinar el valor definitivo que será usado en las corridas posteriores. La forma en la que se hará lo anterior se muestra en la Figura 3.4: se trata de calcular i y s , los valores extremos del intervalo de búsqueda incluyendo a p .

Figura 3.4 Generación de intervalos en ajuste de parámetros



De este modo tenemos que:

$$L = S - I$$

$$l = s - i$$

$$l = rL$$

Donde:

p : Mejor valor encontrado en la fase de valores extremos

I : Valor extremo inferior

S : Valor extremo superior

i : Valor inferior del intervalo de búsqueda exhaustiva

s : Valor superior del intervalo de búsqueda exhaustiva

L : Longitud del intervalo de extremos

l : Longitud del intervalo de búsqueda exhaustiva

$r \in (0,1)$: Tasa de proporcionalidad de delimitación¹²

Se puede observar que las distancias de p a los límites inferior y superior del intervalo no necesariamente son iguales, esto debido a que, a veces, es prioritario considerar valores cargados a un lado u otro de éste, e.g. una tasa de mutación podría adoptar valores muy cercanos a cero, como $p = 0.001$. Supóngase que se calcula un $l = 0.03$, entonces para que p sea el valor central, el intervalo debería ser $\left[0.001 - \frac{0.03}{2}, 0.001 + \frac{0.03}{2}\right] = [-0.014, 0.016]$, que incluye valores no válidos para tal parámetro. Será pues ventajoso desplazar el intervalo a la derecha hasta que el límite inferior sea válido y eventualmente, hasta posiblemente igual a p .

El análisis de la presente fase se hará incluyendo la información de la anterior para evitar cualquier sesgo o información desechada en decisiones previas. Es decir, todas las corridas de la fase 1 se incluirán nuevamente en el

¹² El porcentaje de la longitud del intervalo de extremos que representa la longitud del intervalo de búsqueda exhaustiva, es definido bajo el criterio del investigador de acuerdo al problema.

análisis estadístico comparativo de la fase 2 para evitar que posibles buenos valores de los parámetros hayan sido descartados en la fase 1¹³

3.3.3 Las decisiones de desempeño en el entorno multiobjetivo

Ya fueron mencionadas en la sección 3.1 las métricas propuestas, y de hecho ya se ha señalado que la evaluación de la salida de un algoritmo de optimización multi-objetivo es, en sí, un problema multi-criterio, por lo que no puede usarse una sola métrica. Refiriéndose al problema de ajuste de parámetros, pues bien, si se tienen múltiples criterios, es necesario establecer las que se tomarán en cuenta para evaluar las diferentes configuraciones.

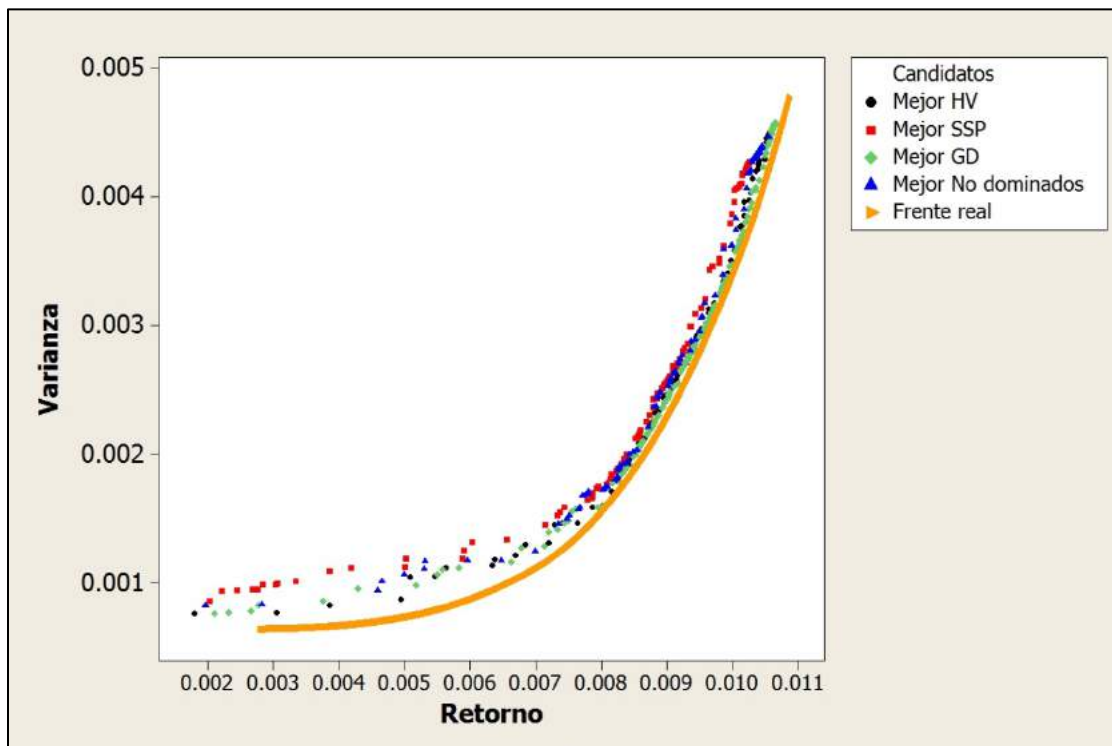
Considere el siguiente experimento, se hace una implementación mediante diseño factorial completo, para poder determinar un juego de parámetros adecuado para el MOEA/D en la instancia más pequeña del problema del portafolio¹⁴. Se obtienen resultados para tres métricas de gran importancia: gd , ESS y HV se podría decir que el mejor resultado obtenido con respecto a una sola métrica es un candidato a mejor resultado en general, dado que posee el mejor valor en al menos una de las tres cualidades. Además de la anteriores candidatas se considerará un criterio más de la siguiente manera: imagine que se tiene un MOP de la forma $\min \mathbf{h}(\mathbf{z}) = (gd(\mathbf{z}), ESS(\mathbf{z}), -HV(\mathbf{z}))$ cuyo espacio factible son los valores para \mathbf{h} obtenidos en el experimento, el frente de Pareto sería un conjunto no dominado de puntos pero en el espacio de las métricas que llamaremos Θ , para seleccionar del anterior conjunto la mejor solución,

¹³ Tome en cuenta que el costo computacional yace en las corridas realizadas y no en la cantidad de resultados que se incluyen en el análisis estadístico.

¹⁴ Diez experimentos por cada juego de parámetros, se calculó el promedio en cada métrica.

démosle mayor prioridad al hipervolumen para ello, así que seleccionemos la solución con mejor valor en ésta métrica y ésta será la mejor solución bajo el cuatro criterio. Para las cuatro mejores soluciones descritas se muestran los frentes de Pareto (en el espacio de los objetivos del problema) en la Figura 3.5

Figura 3.5 Experimento: decisión de desempeño



Nótese que bajo el cuarto criterio es elegido un conjunto de soluciones no dominadas en el espacio de las métricas (Θ) para graficar, por lo cual la decisión de desempate entre tales soluciones, igualmente buenas desde el enfoque de dominancia, es la que tenga el mejor valor para el *HV*.

En la Figura 3.5 se observa que visualmente la solución con mejor *HV* tiene mejores rasgos que las mejores bajo los criterios *gd* y *ESS*¹⁵, ya que es una métrica que mide cualidades de dispersión y convergencia a la vez.

En los experimentos posteriores las decisiones de desempeño no serán tomadas siempre bajo el mismo criterio, en cada etapa se dará su debida justificación, pero, en general será por cuestiones de utilidad de la métrica en el experimento, costo computacional sumamente elevado o por la dificultad de resumir datos de demasiados análisis estadísticos.

3.4 Criterio de paro y análisis del tamaño de las instancias

Uno de los aspectos esenciales de un método heurístico es la decisión de cuándo detener el algoritmo y recuperar los resultados finales, esta decisión puede formularse en función de factores como tiempo de ejecución permitido, número de iteraciones permitidas, una meta a alcanzar para alguna métrica, etc. Trabajar con Algoritmos Evolutivos Multiobjetivo (MOEAs por sus siglas en inglés) posibilita usar como criterio de paro el número de generaciones, pero, debido a que no todos los que fueron probados efectúan el mismo esfuerzo para crear una nueva generación, será más pertinente elegir el número de llamadas o evaluaciones en la función objetivo como criterio para los

¹⁵ Con esto nos estamos refiriendo a que, para hacer una estimación de la curva ésta se mostraría más amplia y parecida a la real, no hay que perder de vista que al hacer la selección de no dominadas en el espacio de las métricas, ya hemos considerado las otras métricas de cierto modo

experimentos realizados en esta investigación. Cabe mencionar que es también el que generalmente se utiliza en el estado del arte¹⁶.

Por otro lado se espera lógicamente que a problemas más complejos les sea asignada una cantidad mayor de recursos (evaluaciones, tiempo CPU). Esta complejidad se relaciona con el tamaño de la instancia tratada. En optimización, características como el número de variables, el tamaño del espacio factible, el tipo de función y el número de objetivos pueden definir el tamaño del problema. Así pues, hablar del concepto tamaño es un tanto relativo, por tal motivo en esta sección se definirá como el número de variables que tiene el modelo.

De acuerdo al modelo matemático de la presente investigación presentado en el apartado 1.1.1, las variables las representa el conjunto de N activos posibles. De este modo N definirá el tamaño de la instancia y de acuerdo a la Tabla 1.1 serán 5 diferentes tamaños utilizados.

Como ya se mencionó, para instancias de mayor tamaño, será necesaria una cantidad mayor de recursos. En la Figura 3.6 se muestra el comportamiento de las métricas de desempeño al aplicar un algoritmo a instancias de diferentes tamaños, pero trabajando con el mismo número de llamadas a la función objetivo¹⁷. Se puede observar que la calidad decae a medida que el tamaño aumenta (GD y ESS aumentan, mientras que $card$ y HV_{perc} disminuyen). Se

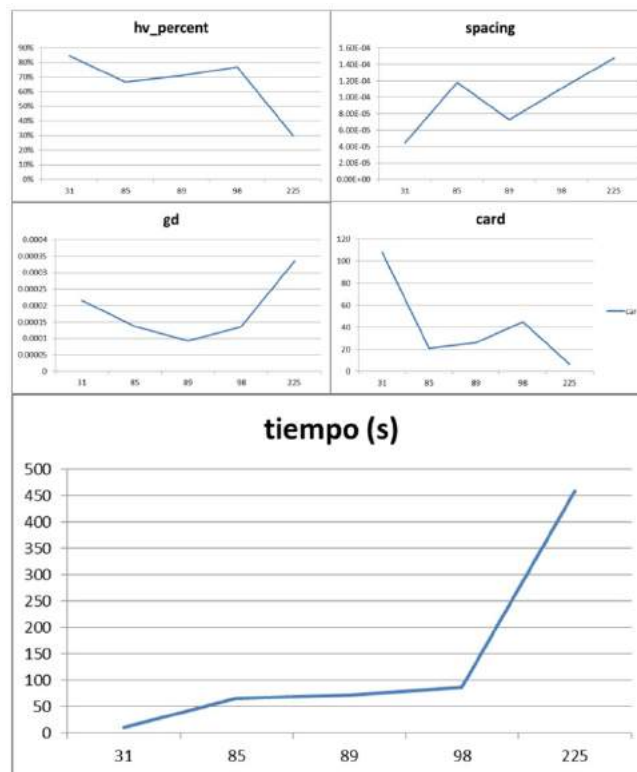
¹⁶ Es preciso aclarar que al definir un criterio de paro, se trata de dejar las condiciones lo más similar posible entre algoritmos probados para que se pueda hacer una comparación lo más justo posible. Por otro lado hay que tomar en cuenta que los algoritmos suelen ser muy distintos en muchos aspectos de implementación (ver capítulo 2), por lo que hablar de “condiciones iguales” resultaría prácticamente inverosímil.

¹⁷ Algoritmo basado en una suma ponderada de objetivos, 15,000 llamadas a la función objetivo, tasa de mutación 0.05, tamaño de población 30, $\lambda = 21$ y generaciones totales 24

entiende que, a medida que el tamaño de la instancia crece, se le debe permitir una ejecución mayor para que los resultados alcancen buenos niveles de convergencia. Nótese incluso que instancias más grandes implican una evaluación de la función objetivo mucho más costosa, causando el crecimiento del tiempo CPU mostrado en la Figura 3.6.

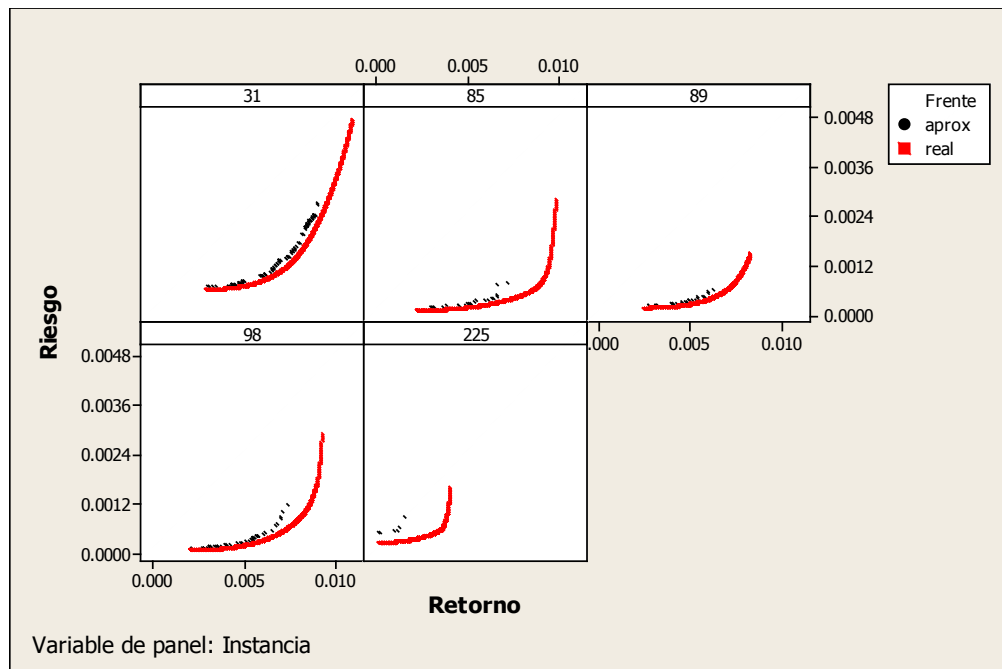
Asimismo, en la Figura 3.7 se puede apreciar que para la instancia más grande, la aproximación gráfica no alcanza buenos valores en ninguno de los rasgos calculados. Usualmente es conveniente proponer una función $g(N)$ que asigne un número total de iteraciones (criterio de paro) permitidas para resolver una instancia de tamaño N .

Figura 3.6 Métricas (ordenadas) respecto al tamaño de la instancia (abscisas)



De igual modo, en la Figura 3.6 es corroborado que, pese a que el número de llamadas a la función objetivo se mantiene constante, el tiempo es creciente, es decir, no solo las llamadas a la función objetivo influyen en el costo computacional.

Figura 3.7 Frentes aproximados respecto al tamaño de la instancia



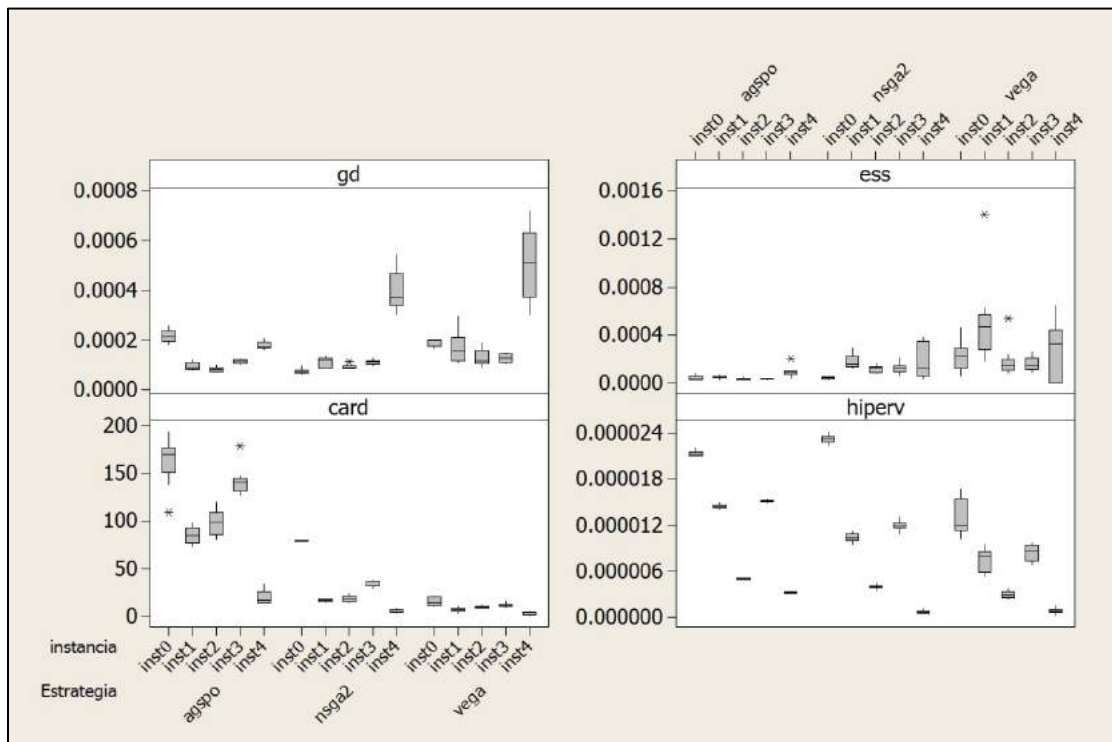
A continuación se muestra otro ensayo en el que se han probado tres de las técnicas presentadas en esta tesis: NSGA-II, Algoritmo Genético basado en una Suma Ponderada De Objetivos (SPO) y VEGA. Fueron realizados 30 ensayos usando 30,000 llamadas a la función objetivo y los parámetros (Tabla 3.2) fueron calibrados de forma exhaustiva.

Tabla 3.2 valores de los parámetros en el ejemplo

Estrategia	instancia	tm	Ψ	Gmax
nsga2	inst0	0.1	80	375
Spo	inst0	0.01	80	18
vega	inst0	0.01	100	300

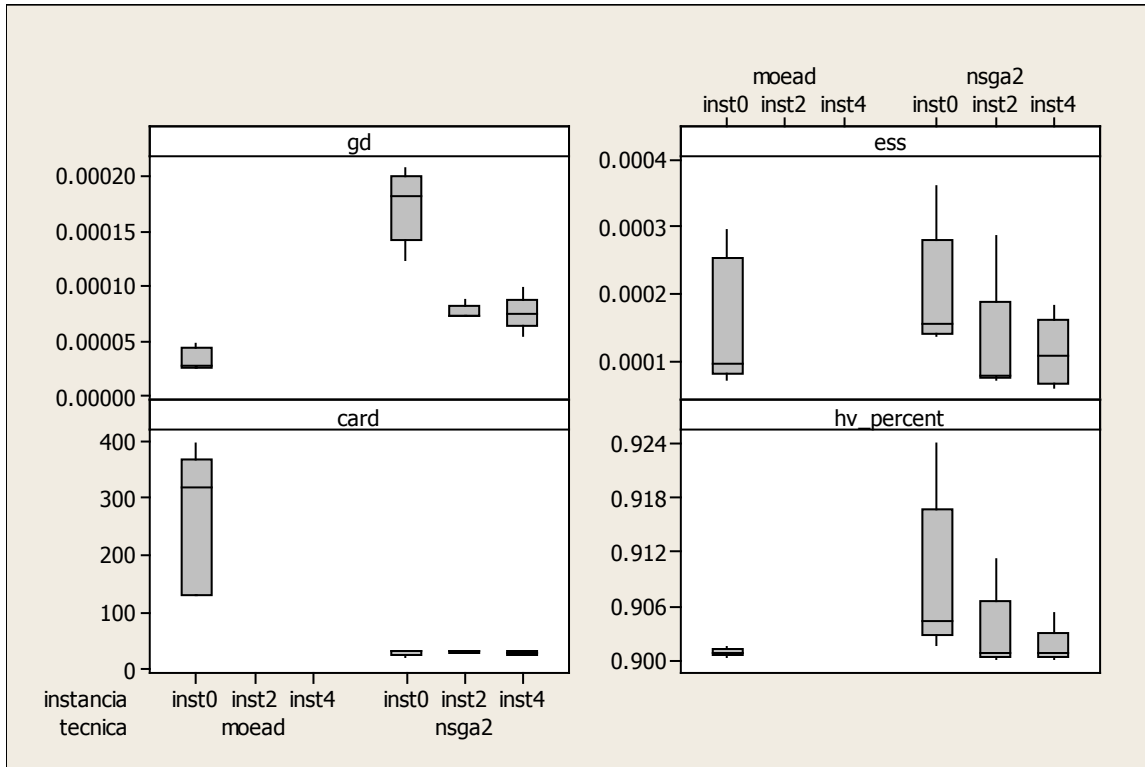
A pesar de que en este ejemplo es posible distinguir las mismas tendencias de deterioro de resultados a causa del tamaño, parece ser que además esto no es lo único relevante: cada técnica tendrá un comportamiento muy particular ante el escenario que afronte. SPO generó frentes más uniformes (*ESS*), NSGA-II los de mejor convergencia (*GD*) y parece ser que, de acuerdo al hipervolumen, la instancia de tamaño 98 tiene alguna característica en particular que la hace más sencilla de atacar (si se observa nuevamente la Figura 3.7 tal vez esto se deba a lo uniforme que es la convexidad del frente).

Figura 3.8 Comportamiento de métricas al tamaño de la instancia



3.4.1 Pruebas de alcance y la función para el criterio de paro

Se justificó en los párrafos anteriores la necesidad de proponer una función $g(N)$ para fijar un criterio de paro para los algoritmos. Para ello, se presentará una serie de pruebas realizadas para tratar de estimar tal función o, en su defecto, proponer la más ad hoc. El experimento, denotado prueba de alcance, consiste en hacer una serie de corridas en las cuales se toma como criterio de paro una meta en particular, en este caso se trata de un cierto valor para el HV_{perc} . Al determinar el número de generaciones en el que se alcanza tal valor, se obtiene una correspondencia entre el tamaño de las instancias y el número de llamadas a la función objetivo que se debería permitir, para garantizar una aproximación del frente que cumpla la meta en términos del hipervolumen. Primero, se intentó con un valor meta para la métrica de $HV_{perc} = 90\%$ (es decir, el frente aproximado cubre un 90% del hipervolumen cubierto por el frente real) Se usaron para estos experimentos las estrategias NSGA-II y MOEA/D, las dos técnicas de mayor interés (por ser las más recientes) en la presente investigación, para resolver las instancias de tamaños 31, 89 y 225 los resultados obtenidos para las diferentes métricas se muestran en la Figura 3.9.

Figura 3.9 Métricas, prueba de alcance $HV_{perc} = 90\%$ 

Observe que no hay resultados en la prueba para el MOEA/D más que para inst0, esto se debe a que tal algoritmo nunca fue capaz de alcanzar la meta propuesta ($HV_{perc} = 90\%$) en un número de generaciones razonable. Por otro lado también se grafica en la Figura 3.10 el número de llamadas a la función objetivo requerido para alcanzar la meta. A pesar de tener información para una sola instancia (inst0), es visiblemente claro que MOEA/D necesita un mucho mayor número de llamadas a la función objetivo para lograr el nivel deseado. Más precisamente, el MOEA/D necesita aproximadamente 112 veces ($\alpha_1 = \frac{24,180}{216} = 111.94$) el número de evaluaciones que realiza el NSGA-II (ver Tabla 3.3).

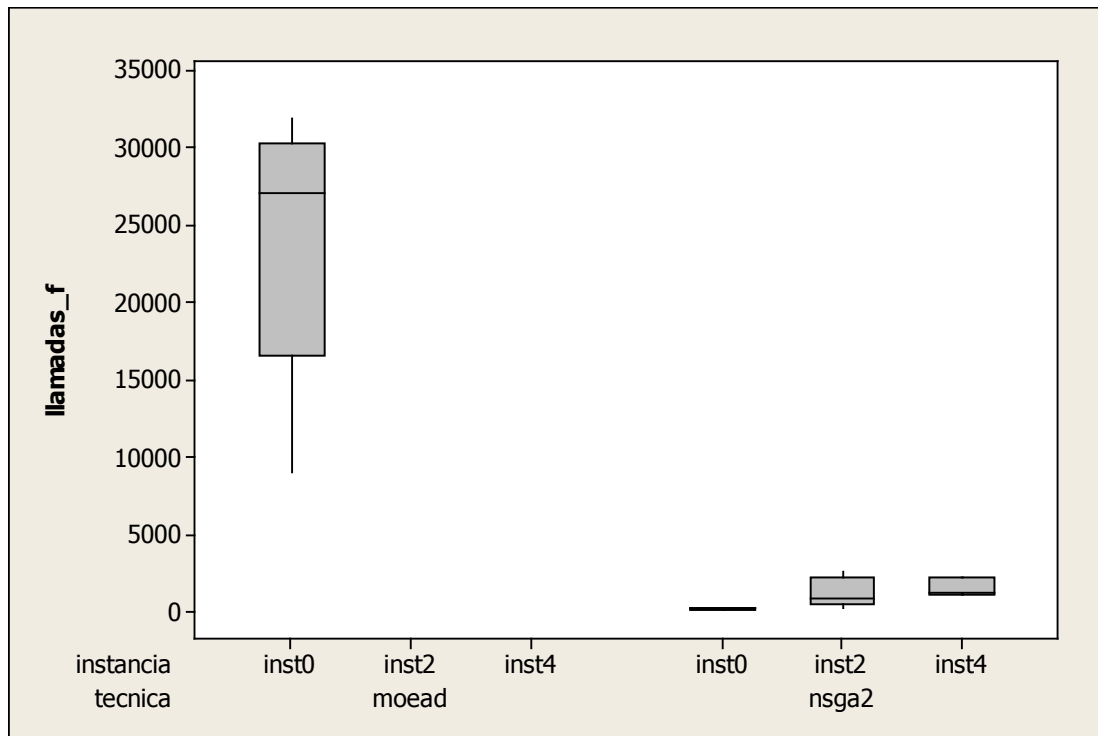
Figura 3.10 Llamadas a la función objetivo, prueba de alcance $HV_{perc} = 90\%$ 

Tabla 3.3 Estadísticos prueba de alcance inst0

tecnica	Media	Desv.Est.	Mediana
moead	24180	8948	27150
nsga2	216	49.3	210

Poniendo por lo pronto a parte el desempeño del MOEA/D (que se comentará más adelante), algunos aspectos clave pueden ser despejados por ahora:

- NSGA-II ya muestra una gran eficiencia para resolver el problema,
- El valor para HV_{perc} donde el algoritmo se detiene va disminuyendo con el aumento en el tamaño de la instancia (se va acercando a 90% por la derecha) de acuerdo a esto se puede conjeturar que, con el aumento en el

tamaño de la instancia, generación tras generación, el progreso es más lento al menos al final de la corrida¹⁸,

- Con *GD* y *ESS* sucede una tendencia opuesta a la anterior, lo que significa que al alcanzar un 90% del *HV* real, ya son mejores los frentes para las instancias más grandes que para las pequeñas (esto claro, con respecto a los criterios *GD* y *ESS*).

Con respecto al comportamiento del MOEA/D ya se mencionó que no le fue posible alcanzar la meta a partir de inst2, mientras que en la Tabla 3.4 se puede ver que NSGA-II necesita en promedio 42 generaciones para alcanzar el nivel requerido $HV_{perc} = 90\%$. En el primer renglón de la Tabla 3.4 se presentan también los valores de las otras métricas obtenidas cuando ha alcanzado $HV_{perc} = 90\%$, en el segundo renglón en cambio, se indica que el MOEA/D necesita 5,000 generaciones para alcanzar una eficiencia similar al NSGA-II respecto a *ESS* y 25,000 generaciones para lograr la misma convergencia (*GD*).

Tabla 3.4 Métricas, alcance promedio inst2

	Gmax	gd	ess
valores promedio en los que nsga2 alcanza el 90% de hv_percent	41.8	7.6613E-05	0.00012046
número de generaciones que MOEA/D necesitó para alcanzar el valor promedio	-	25,000	5,000

¹⁸ El algoritmo no se detiene justo cuando la métrica es igual a 90% sino cuando la métrica es igual o mayor (se tiene cubierta la meta): puede suceder que en la generación j sea 89.92%, posteriormente la etapa $j + 1$ sea muy fructífera y se llegue a 93%. Es por eso que se incluye también HV_{perc} en el gráfico: la diferencia entre el valor alcanzado y 90% puede darnos una idea de a qué velocidad mejoraba la métrica en ese instante.

Si bien no fue posible alcanzar la meta en hipervolumen con el MOEA/D, la Tabla 3.4 permite afirmar que al menos serán necesarias aproximadamente ($\alpha_2 = \frac{5,000}{41.8} = 119.62$) 120 veces el número de generaciones del NSGA-II para que el MOEA/D empate a NSGA-II en términos de *ESS*.

Se han calculado dos proporciones de eficiencia entre los dos algoritmos probados (α_1 y α_2), las cuales serán usadas para fijar un tope a las llamadas a la función objetivo. De acuerdo a lo anterior, en promedio MOEA/D necesitará $\bar{\alpha} = \frac{\alpha_1 + \alpha_2}{2} = \frac{111.94 + 119.62}{2} = 115.78$ veces los recursos ocupados por el NSGA-II, así que se usará posteriormente esta proporción para aproximar a $g(N)$.

Por medio de una regresión lineal, se hizo una estimación (la cual fue la que mejor se ajustó) de la función $g(N)$, ilustrada en la Figura 3.11. Cabe mencionar que esta regresión fue considerando solamente la información obtenida con el NSGA-II. Al tomar en cuenta además la proporción $\bar{\alpha}$ (factor de corrección para el MOEA/D), entonces se puede definir:

$$g(N) = \bar{\alpha}g(N) = 115.78(8.0864N^{0.9955}) = 936.24N^{0.9955} < 1000N$$

$$\text{para } N \geq 1$$

Basado en el análisis anterior, se decidió establecer un límite de llamadas a la función objetivo o criterio de paro como $g(N) = 1000N$, dado que cubre ligeramente por encima a la obtenida con las estimaciones anteriores (Figura

3.12). Se espera de esta manera obtener datos lo suficientemente buenos para hacer comparaciones entre los algoritmos implementados.

Figura 3.11 Estimación de función para el criterio de paro (NSGA-II)

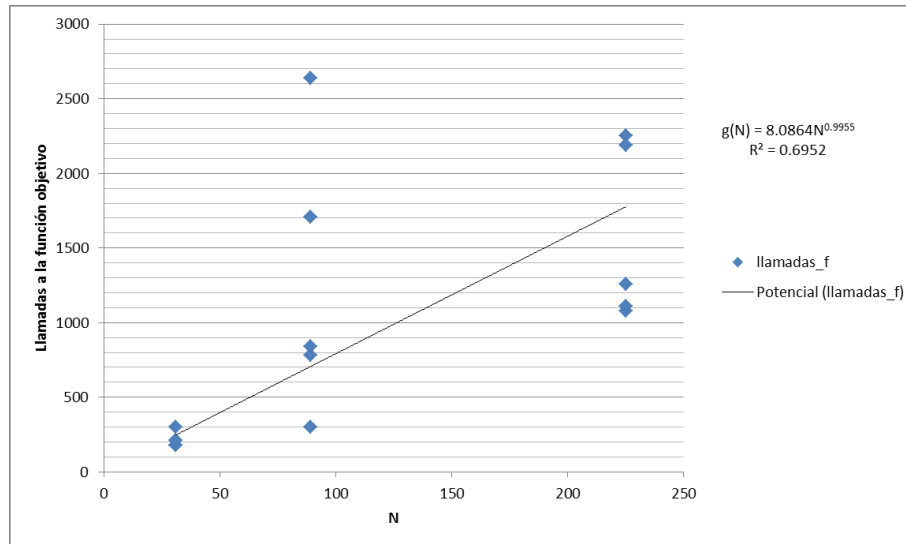
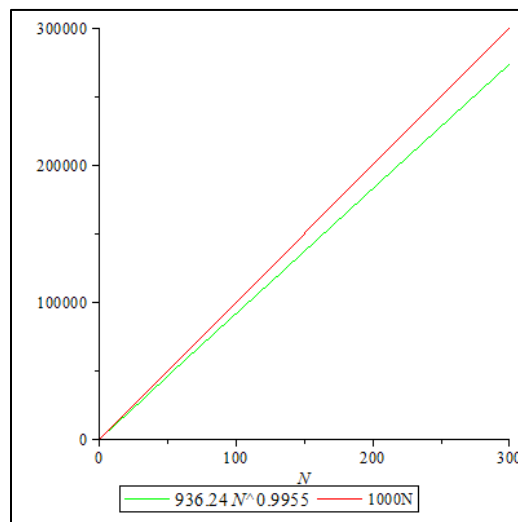


Figura 3.12 Función $g(N)$ estimada (verde) y propuesta (rojo)



3.5 Estrategias de selección, cruza y mutación

Debido a que se trata de tres aspectos de gran relevancia en el funcionamiento de un AG, fue necesario realizar un conjunto de experimentos

para decidir cuáles son los métodos adecuados para optimizar la eficiencia de las metaheurísticas. En este apartado se presentan los resultados obtenidos y las conclusiones correspondientes.

3.5.1 Selección

Basado en la premisa de la supervivencia del más apto (ver sección 1.3.2), este proceso relacionado con la presión ejercida hacia buenas soluciones, puede influenciar el comportamiento elitista (o no) del algoritmo. Define bajo qué criterios los individuos serán involucrados para compartir información en la fase de cruce y para sobrevivir. En el estado del arte se han propuesto diversos métodos entre los cuales están: selección por la regla de la ruleta, selección por ranking, selección por torneo binario, selección estocástica universal, criterio completamente elitista (los mejores individuos son escogidos), etc.

Para la presente investigación, se utilizó una selección por torneo binario en todas las estrategias implementadas, sea para la selección ambiental o la selección de padres; el Algoritmo 3.1 describe el procedimiento correspondiente. La razón por la que se decidió utilizar tal método es evitar el sesgo debido al número de copias esperadas, presente en varias opciones probabilísticas, como la ruleta por ejemplo. Estas técnicas no se mostrarán tan satisfactorias cuando existen grandes diferencias entre las aptitudes de los individuos de la población dado que si por tener una aptitud mucho mayor a los demás, un cromosoma ocupa gran parte de la ruleta (o espectro de probabilidad correspondiente), el

resto de ellos tienen muy pocas posibilidades de ser elegidos y la pérdida de diversidad asociada puede deteriorar fuertemente el desempeño del algoritmo.

Cabe mencionar que al tratarse de optimización multiobjetivo, para definir la aptitud de los individuos, se utiliza un criterio de dominancia de Pareto, el cual más que un valor numérico, es un estado (dominado o no dominado), lo cual es más sencillo hacer análogo a un torneo.

En el caso de que los individuos en el torneo sean no dominados entre sí la selección del ganador se hace de forma aleatoria (Algoritmo 3.1).

Algoritmo 3.1 Selección por torneo binario (minimizar)

```
padres = ∅
para i en [0, 1] hacer:
    p1 = escoger elemento aleatorio en población
    gana = ∅
    mientras gana == ∅:
        p2 = escoger elemento aleatorio en población
        fit_1 = fitness de p1
        fit_2 = fitness de p2
        si fit_1 < fit_2:
            gana = p1
        de otro modo si fit_1 < fit_2:
            gana = p2
        de otro modo:
            gana = escoger elemento aleatorio en {p1, p2}
    agregar gana a padres
```

3.5.2 Cruza y Mutación

La cruce es la etapa en la cual soluciones de la generación actual (padres) comparten su patrimonio genético, es decir, que recombinan sus características

propias para generar nuevas soluciones llamadas hijos, las cuales poseerán rasgos heredados de sus padres.

Por otro lado, los algoritmos suelen, bajo una cierta probabilidad, introducir perturbaciones locales en los individuos con el objetivo de diversificar la población, provocando desviaciones para evitar la convergencia prematura del proceso de búsqueda. A este proceso se le conoce como mutación.

Se experimentó utilizando dos metodologías de cruce y mutación, adaptadas para la codificación real de las variables: uniforme y, por otro lado, Simulated Binary Crossover con su mutación paramétrica asociada.

Cruza y mutación uniforme

Consiste en realizar las operaciones genéticas de tal forma que su comportamiento siga una distribución uniforme.

La cruce se realiza de la siguiente manera, sean $p_1 = \langle a_1, a_2, \dots, a_n \rangle$ y $p_2 = \langle b_1, b_2, \dots, b_n \rangle$ los padres, entonces:

$$h_1 = \langle z_c a_1 + (1 - z_c) b_1, z_c a_2 + (1 - z_c) b_2, \dots, z_c a_n + (1 - z_c) b_n \rangle$$

$$h_2 = \langle (1 - z_c) a_1 + z_c b_1, (1 - z_c) a_2 + z_c b_2, \dots, (1 - z_c) a_n + z_c b_n \rangle$$

$$z_c \in [0,1]$$

Donde h_1 y h_2 son los hijos producidos, n es el número de variables y z_c es un elemento aleatorio entre 0 y 1 que sigue una distribución uniforme. Una

ventaja de este método de cruce es que no hay que precisar algún parámetro para su implementación.

Por otro lado, para la mutación es necesario establecer un valor: la tasa de mutación (z_m), que representa la probabilidad de mutar una variable. Para mutar un individuo bastará con generar para cada alelo un número aleatorio u , si $u < z_m$ entonces se aplica una perturbación al alelo, de lo contrario no se realiza ningún cambio. Tal perturbación se trata de una regeneración aleatoria (también uniforme) del valor del alelo en el dominio de la variable que le corresponde.

Simulated Binary Crossover (SBX)

Se trata de un método propuesto por Deb & Bhushan Agrawal (1995) el cual intenta emular el efecto de la cruce de un punto, usado en una representación binaria. La forma en la que se implementa es la siguiente:

1. Sean $p_1 = \langle a_1, a_2, \dots, a_n \rangle$ y $p_2 = \langle b_1, b_2, \dots, b_n \rangle$ los padres, generar un número aleatorio $u \in [0,1]$
2. Calcular el parámetro $\bar{\beta}$ de la siguiente manera¹⁹:

$$\bar{\beta} = \begin{cases} (2u)^{\frac{1}{n_c+1}} & \text{si } u \leq 0.5 \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{n_c+1}} & \text{de lo contrario} \end{cases}$$

3. Producir hijos de la siguiente forma:

¹⁹ Los autores sugieren $n_c \in \{1,2\}$

$$h_1 = \langle 0.5[(a_1 + b_1) - \bar{\beta}|b_1 - a_1|], 0.5[(a_2 + b_2) - \bar{\beta}|b_2 - a_2|], \dots, 0.5[(a_n + b_n) - \bar{\beta}|b_n - a_n|] \rangle$$

$$h_2 = \langle 0.5[(a_1 + b_1) + \bar{\beta}|b_1 - a_1|], 0.5[(a_2 + b_2) + \bar{\beta}|b_2 - a_2|], \dots, 0.5[(a_n + b_n) + \bar{\beta}|b_n - a_n|] \rangle$$

En conjunto con SBX, se ha implementado la mutación paramétrica propuesta por los mismos autores (Deb & Bhushan Agrawal, 1995), la cual aplica el siguiente procedimiento:

1. Dada una solución $h = \langle a_1, a_2, \dots, a_n \rangle$, para cualquier posición j a mutar, generar un número aleatorio $u \in [0,1]$ con distribución uniforme
2. Hacer el cálculo de $\bar{\delta}$ de la siguiente manera:

$$\bar{\delta} = \begin{cases} (2u)^{\frac{1}{\eta_m+1}} & \text{si } u \leq 0.5 \\ UB & \text{de lo contrario} \end{cases}$$

Donde η_m es el índice de distribución para la mutación y toma cualquier valor no negativo²⁰

3. El valor de la posición j se determina usando:

$$a'_j = a_j + \bar{\delta}\Delta_{max}$$

Donde Δ_{max} es la máxima perturbación permitida, usualmente se utiliza $\Delta_{max} = UB - LB$, donde UB es el límite superior y LB el límite inferior de la variable ($a_j \in [LB, UB]$).

Nótese que, al igual que en las cruas y mutación uniformes, no es necesario establecer un parámetro, esto solamente si se toman las sugerencias de los autores, las cuales no garantizan el funcionamiento eficiente en toda instancia o

²⁰ Los autores sugieren $\eta_m = 100 + t$, donde t es la generación actual.

problema, por lo tanto, una desventaja de la técnica puede ser el control de dichos parámetros.

3.5.3 Análisis comparativo

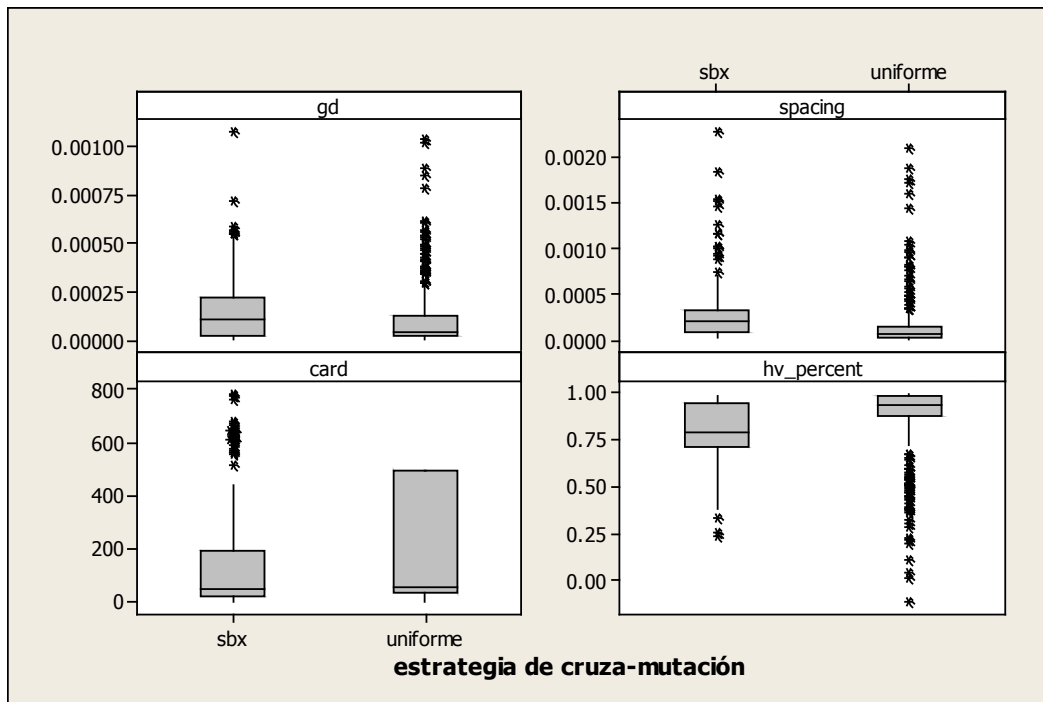
Antes de mostrar resultados, se debe recordar que se han trabajado cinco instancias de diferentes tamaños (Tabla 1.1) y cuatro estrategias de optimización multiobjetivo; también hay que tomar en cuenta que la calidad de las soluciones es evaluada con respecto a múltiples métricas. En el mejor de los casos, se espera determinar métodos de cruce y mutación que puedan ser usados en todas las técnicas por igual, pero esto no será posible.

A continuación se muestra el experimento realizado que consiste en una implementación de las metaheurísticas usando las dos técnicas de cruce y mutación. Las características de implementación son las siguientes:

- Diez experimentos por cada técnica de cruce y mutación probada en cada metaheurística.
- Análisis previo de ajuste de parámetros de acuerdo a la sección 3.3
- Manejo de restricciones por reparación aleatoria (ver sección 3.6.2)
- Criterio de paro: $1000N$ llamadas a la función objetivo, donde N es el número total de activos de la instancia de acuerdo al modelo matemático.
- Archivo EP en el MOEA/D restringido a Ψ elementos, donde Ψ es el tamaño de la población.

De acuerdo a los resultados obtenidos para todas las metaheurísticas, los cuales se muestran en la Figura 3.13, utilizando como criterio la mediana, se podría afirmar que para todas las métricas fue mejor la técnica uniforme, pero la presencia de una gran cantidad de datos atípicos sugiere la necesidad de un análisis más detallado.

Figura 3.13 Métodos de cruce-mutación (a) todas las estrategias



Si se separa en cambio los resultados con respecto a cada estrategia de acuerdo a la Figura 3.14 se observa que, considerando todas las métricas, para las estrategias NSGA-II y MOEA/D funciona mejor el método uniforme, mientras que para SPO se obtiene un mejor desempeño usando SBX²¹. En cuanto a

²¹ Es posible hacer esa conclusión dado que es notorio en el gráfico la diferencia no solo en la medida central Mediana, sino también en la forma en la que se distribuyen los datos (los intervalos que cubren las cajas son excluyentes)

VEGA si se observa la información como se muestra en la Figura 3.15, para la métrica de mayor relevancia (hipervolumen) y la cardinalidad la opción es SBX.

Figura 3.14 Métodos de cruce-mutación (b)

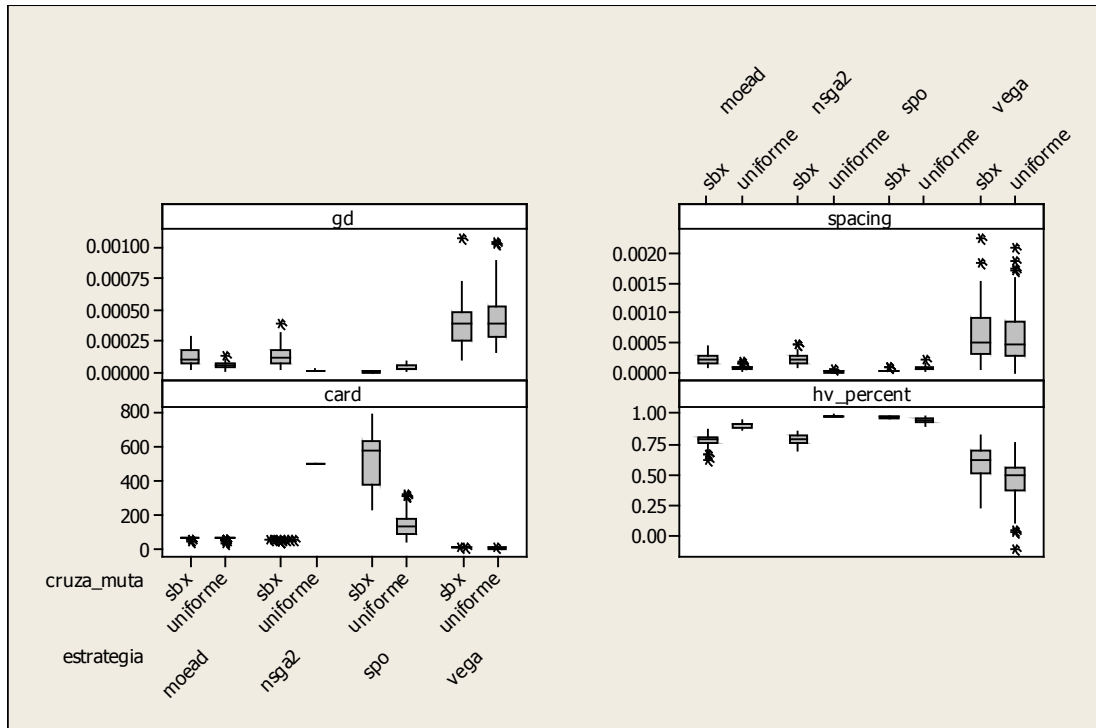
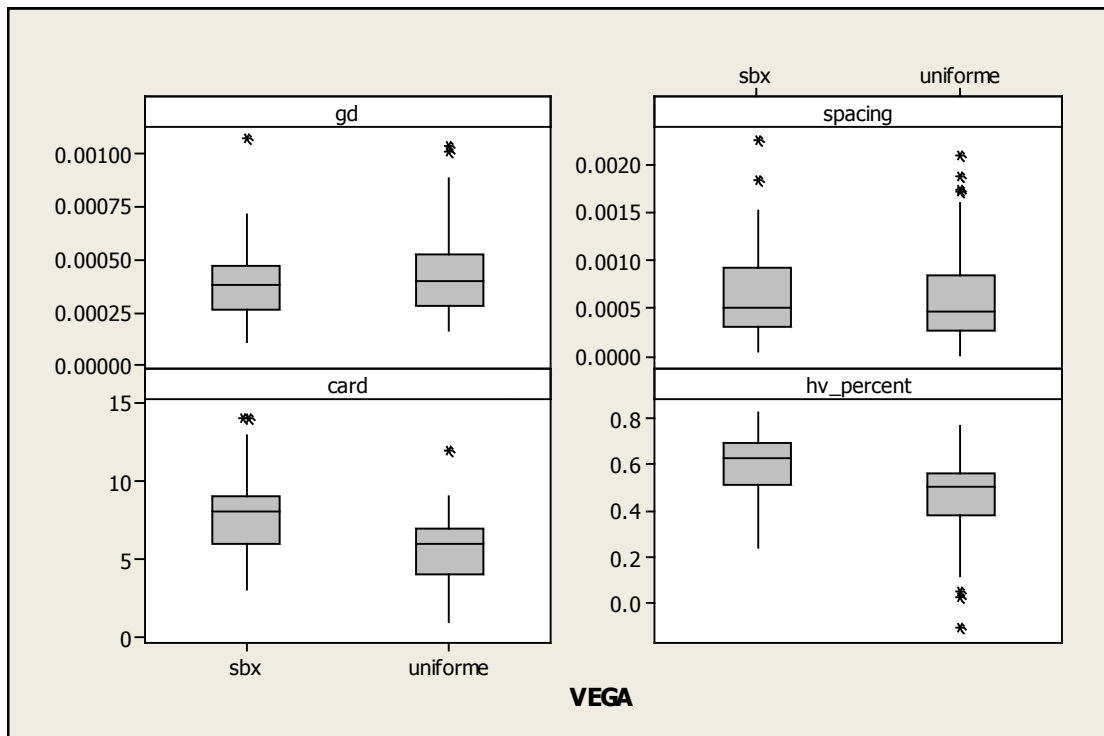


Figura 3.15 Métodos de cruce-mutación VEGA



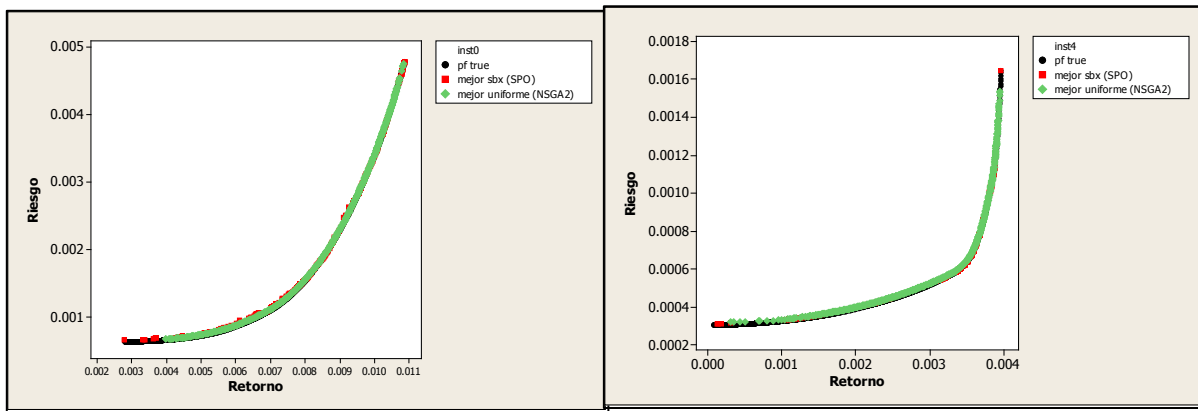
De acuerdo a los resultados obtenidos, dado que cada algoritmo responde de distinta manera a los operadores de cruce-mutación, es preciso que cada metaheurística trabaje con el método con el cuál mostró mejor desempeño.

En la Tabla 3.5 se muestran las métricas de los mejores resultados (de acuerdo al hipervolumen) obtenidos con cada método de cruce y mutación, para las instancias más pequeña y más grande. En la Figura 3.16 se pueden ver los respectivos frentes de Pareto aproximados.

Tabla 3.5 Mejores resultados por método de cruce-mutación (hipervolumen)

estrategia	cruza_muta	instancia	gd	spacing	card	hiperv	tiempo	hv_percent
nsga2	uniforme	inst0	1.2689E-05	7.2335E-06	499	2.5652E-05	331.106526	0.99322262
Spo	sbx	inst0	1.1734E-05	4.3397E-05	325	2.548E-05	9.8659091	0.9865866
nsga2	uniforme	inst4	6.4406E-06	7.9468E-06	500	4.5808E-06	5412.42698	0.99195439
spo	sbx	inst4	1.6338E-06	1.4987E-05	606	4.5346E-06	3692.38968	0.98195126

Figura 3.16 Mejores resultados por método de cruce-mutación (hipervolumen)



3.6 Metodología para el manejo de restricciones

Existen diversas formas de generar una población inicial de soluciones para la implementación de una metaheurística. En la presente tesis siempre se inicia con una población generada aleatoriamente de manera uniforme en el espacio de las variables. Es conocido el intervalo de las variables ($x_i \in [0,1] \forall i = 1, \dots, n$) pero esto no avala el cumplimiento de las restricciones del modelo, por lo tanto, de inicio no tenemos garantía de que todos los individuos de la población inicial sean factibles. En el transcurso del algoritmo se aplican operadores genéticos que modifican las soluciones, que aun siendo factibles pueden dejar de serlo después de las operaciones de cruce y mutación. Una estrategia de manejo de

restricciones permite manejar soluciones no factibles, de tal forma que la mayoría de éstas se mantengan en espacio factible o lo más cercano posible. Muchas veces algunas soluciones, a pesar de no ser factibles, muestran una buena aptitud y vale la pena conservarlas en la población, ya que puede ser indicio de que soluciones muy cercanas (parecidas) también posean tal característica y además sean parte de la región factible.

En la sección 1.1.1 se definió el modelo con el que se probaron las estrategias presentadas en la actual investigación. Es necesario vigilar la restricción que obliga a las variables a sumar estrictamente 1, junto con las condiciones de no negatividad, para mantener la población en zona factible: de esto se ocupa la estrategia de manejo de restricciones. Para determinar la técnica de manejo de restricciones adecuada, se han realizado experimentos utilizando diversos métodos para elegir el que se muestre más eficiente.

3.6.1 Normalización

Dada una solución, posiblemente no factible:

$$p = [x_1, x_2, \dots, x_n]$$

Se define:

$$\alpha = \sum_{i=1}^n x_i$$

La normalización de p será la siguiente:

$$p' = [\frac{x_1}{\alpha}, \frac{x_2}{\alpha}, \dots, \frac{x_n}{\alpha}]$$

De este modo la suma de todos los elementos de p' será siempre igual a 1 (factible).

Ejemplo:

$$\begin{aligned} p &= [0.1, 0.2, 0.5, 0.05] \\ \sum x &= 0.85 \\ p' &= [0.1176, 0.2353, 0.5882, 0.0588] \end{aligned}$$

3.6.2 Reparación aleatoria

La idea de este método, detallado en el Algoritmo 3.2, consiste en, dada una solución p , generar un orden aleatorio s de las variables y crear un nuevo gen $p' = \mathbf{0}$. En el orden s generado anteriormente, se reemplazan los valores de p' por los de p . Al intentar hacer un replazo, hay que comprobar la suma de las x en p' : si el replazo se hiciera y la suma es menor a 1, se continua con la siguiente posición; si la suma fuese igual a 1, se hace el replazo, p' es el gen reparado y el proceso finaliza. Finalmente, si la suma fuera mayor que 1 solo se le asigna a la posición actual el valor necesario para que la suma sea igual a 1. En caso de que, llegando a la posición final, la suma de todas las x fuese menor a 1 aún, la cantidad faltante se le será aumentada a una posición seleccionada aleatoriamente.

Algoritmo 3.2 Método de reparación aleatoria

```

Dada una solución posiblemente no factible  $p$ 
 $p = [x_1, x_2, \dots, x_n]$ 
suma = 0
Hacer  $p' = [y_1, y_2, \dots, y_n] = [0, 0, \dots, 0]$ 
Elegir un orden aleatorio  $s$  de selección de variables
Para cada  $i$  en  $s$ :
    Suma = suma +  $x_i$ 
    Si suma < 1:
         $y_i = x_i$ 
    De otro modo si suma=1:
         $y_i = x_i$ 
        Romper ciclo
    De otro modo:
         $y_i = x_i - (suma - 1)$ 
Si se llegó hasta  $x_n$  y suma<1:
    Seleccionar posición  $j$  aleatoria
     $x_j = x_j + (1 - suma)$ 
Devolver  $p'$ 

```

Ejemplos:

$$\begin{aligned}
 p &= [0.1, 0.2, 0.5, 0.05] \\
 \sum x &= 0.85 \\
 s &= [2, 1, 3, 0] \\
 p' &= [0.25, 0.2, 0.5, 0.05]
 \end{aligned}$$

$$\begin{aligned}
 p &= [0.1, 0.2, 0.5, 0.4] \\
 \sum x &= 1.2 \\
 s &= [2, 3, 1, 0] \\
 p' &= [0.0, 0.1, 0.5, 0.4]
 \end{aligned}$$

$$\begin{aligned}
 p &= [0.1, 0.2, 0.4, 0.3, 0.05, 0.15, 0.25] \\
 \sum x &= 1.45 \\
 s &= [4, 1, 6, 3, 5, 2, 0] \\
 p' &= [0.0, 0.2, 0.05, 0.3, 0.05, 0.15, 0.25]
 \end{aligned}$$

3.6.3 Reparación max-min

Este método funciona de la misma manera que el anterior, excepto que el orden de selección, en vez de aleatorio, se determina de acuerdo a un orden descendiente del valor de las variables en el gen p . La idea es que se mantengan en la solución los activos cuya proporción del capital utilizado es mayor. El proceso se muestra en el Algoritmo 3.3.

Algoritmo 3.3 Método de reparación max-min

```

Dada una solución posiblemente no factible  $p$ 
 $p = [x_1, x_2, \dots, x_n]$ 
suma = 0
Hacer  $p' = [y_1, y_2, \dots, y_n] = [0, 0, \dots, 0]$ 
Elegir un orden  $s$  de selección de las variables de mayor a menor valor para las  $x$  en  $p$ 
Para cada  $i$  en  $s$ :
    Suma = suma +  $x_i$ 
    Si suma < 1:
         $y_i = x_i$ 
    De otro modo si suma=1:
         $y_i = x_i$ 
        Romper ciclo
    De otro modo:
         $y_i = x_i - (suma - 1)$ 
Si se llegó hasta  $x_n$  y suma<1:
    Seleccionar posición  $j$  aleatoria
     $x_j = x_j + (1 - suma)$ 
Devolver  $p'$ 

```

Ejemplos:

$$\begin{aligned}
 p &= [0.1, 0.2, 0.5, 0.05] \\
 \sum x &= 0.85 \\
 s &= [2, 1, 0, 3] \\
 p' &= [0.1, 0.2, 0.5, 0.2]
 \end{aligned}$$

$$\begin{aligned}
 p &= [0.1, 0.2, 0.5, 0.4] \\
 \sum x &= 1.2 \\
 s &= [2, 3, 1, 0] \\
 p' &= [0.0, 0.1, 0.5, 0.4]
 \end{aligned}$$

$$p = [0.1, 0.2, 0.4, 0.3, 0.05, 0.15, 0.25]$$

$$\sum x = 1.45$$

$$s = [2, 3, 6, 1, 5, 0, 4]$$

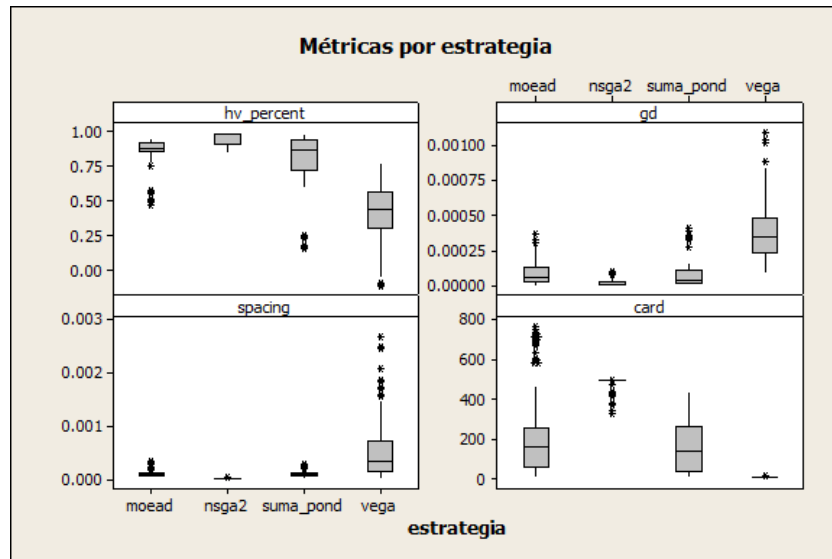
$$p' = [0.0, 0.05, 0.4, 0.3, 0.0, 0.0, 0.25]$$

3.6.4 Análisis comparativo

La etapa de experimentación para seleccionar el método de manejo de restricciones se realizó bajo las siguientes condiciones:

- 30 experimentos por técnica de manejo de restricciones, por metaheurística y por instancia
- Análisis previo de ajuste de parámetros de acuerdo a la sección 3.3
- Cruza y mutación de acuerdo al análisis hecho en la sección 3.5.3, es decir en NSGA-II y MOEA/D se utiliza el método uniforme, mientras que a SPO y a VEGA les corresponde SBX
- Criterio de paro: $1000N$ llamadas a la función objetivo, donde N es el número total de activos de la instancia de acuerdo al modelo matemático.
- Archivo EP en el MOEA/D restringido a Ψ elementos, donde Ψ es el tamaño de la población.

Figura 3.17 Métricas por estrategia



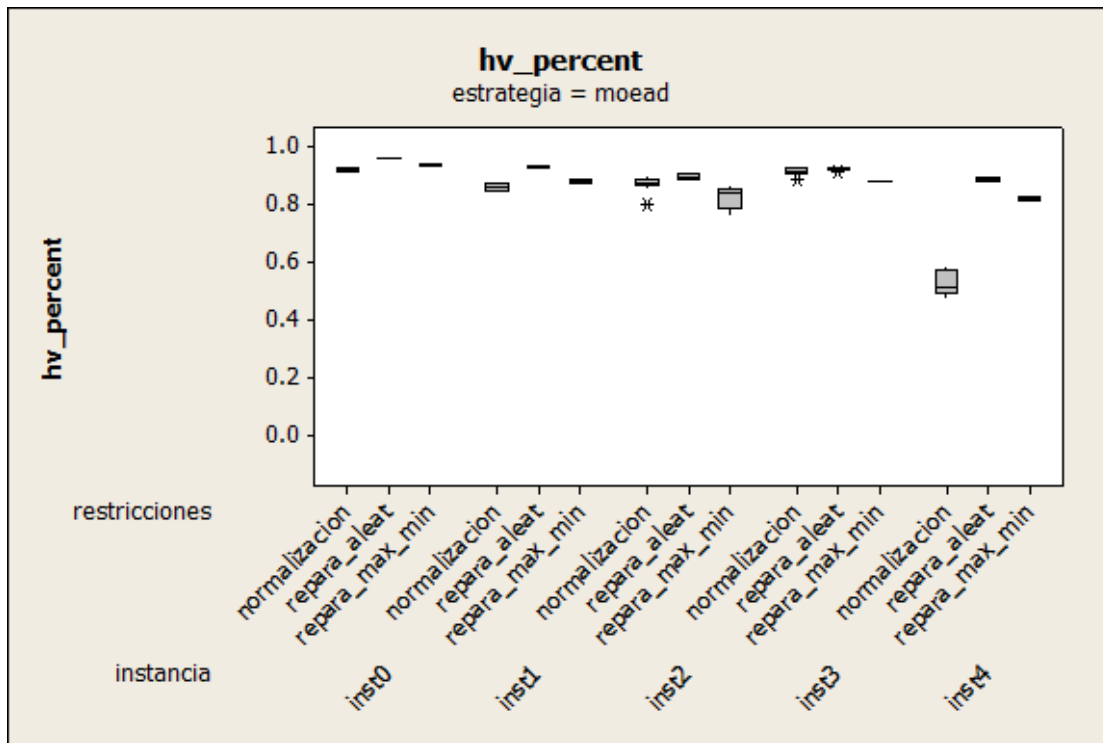
En la etapa experimental de la selección de método de cruce y mutación (3.5.3) se empezó a notar, a grandes rasgos, que los métodos de mejor eficiencia fueron NSGA-II y SPO. En la presente etapa experimental, antes de analizar los métodos de manejo de restricciones, será aprovechada la información obtenida, para hacer algunas conclusiones generales sobre la eficiencia de los algoritmos hasta este punto de la investigación. De acuerdo a la Figura 3.17, la cual valora la eficiencia de cada técnica (sin importar la forma en la que se manejaron las restricciones) el orden de efectividad de mayor a menor es: NSGA2, MOEA/D, SPO y VEGA; Lo anterior con respecto solo a las métricas HV_{perc} , gd y ESS . Las cardinalidades son muy variables con respecto a cada metaheurística y no definen la calidad del frente aproximado²², además, tómese en cuenta que se hace una calibración de parámetros que no suele

²² Por otro lado es importante no perder de vista tal métrica, dado que es posible detectar generalidades de acuerdo a su valor, e.g. valores muy pequeños sugieren una imposibilidad de obtener buen desempeño, a pesar de que otras métricas alcancen valores aceptables. La dispersión (varianza) de la cardinalidad, puede sugerirnos un grado de inestabilidad de la técnica si su valor es alto.

sugerir el mismo valor para el tamaño de la población en todas las técnicas. Hasta ahora, VEGA se ha mostrado como una técnica muy inestable (en todas las métricas muchos valores atípicos, varianza muy alta).

Volviendo al manejo de restricciones, los resultados se presentan de la Figura 3.18 a la Figura 3.21 para cada una de las técnicas, tomando en cuenta el tipo de manejo de restricciones en cada una de las instancias. Se toma en cuenta la métrica de desempeño HV_{perc} .

Figura 3.18 HV_{perc} manejo de restricciones (MOEA/D)



La conclusión respecto a MOEA/D (Figura 3.18) es que la reparación aleatoria es la de mayor eficiencia, para NSGA-II (Figura 3.19) puede afirmarse que normalización tuvo el peor desempeño, de acuerdo a la Figura 3.20 la reparación

aleatoria es la opción para SPO y finalmente, para VEGA, en tres de cinco instancias se obtuvieron mejores resultados con la reparación aleatoria.

Figura 3.19 HV_{perc} manejo de restricciones (NSGA-II)

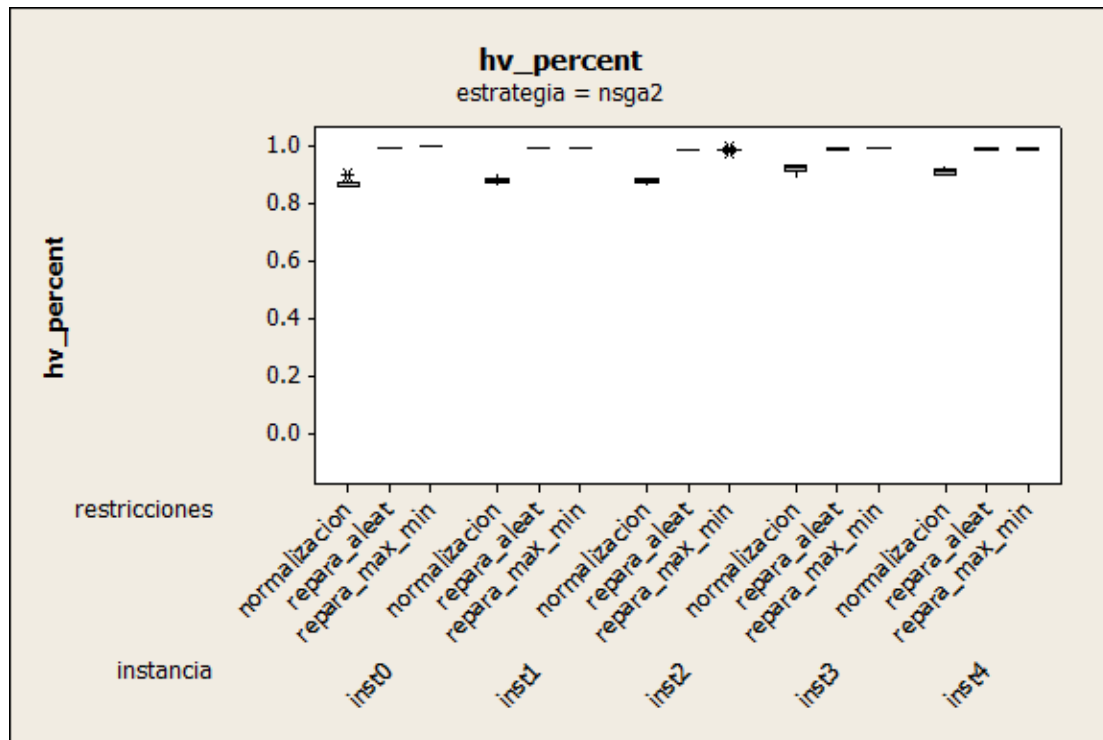


Figura 3.20 HV_{perc} manejo de restricciones (SPO)

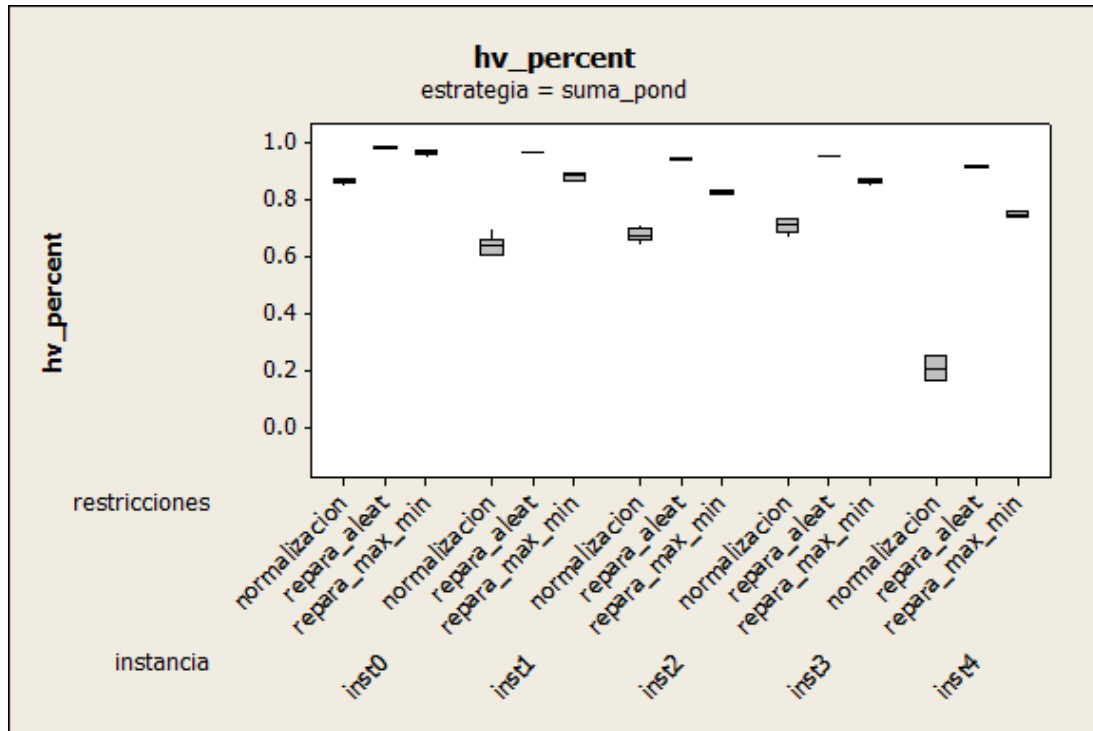
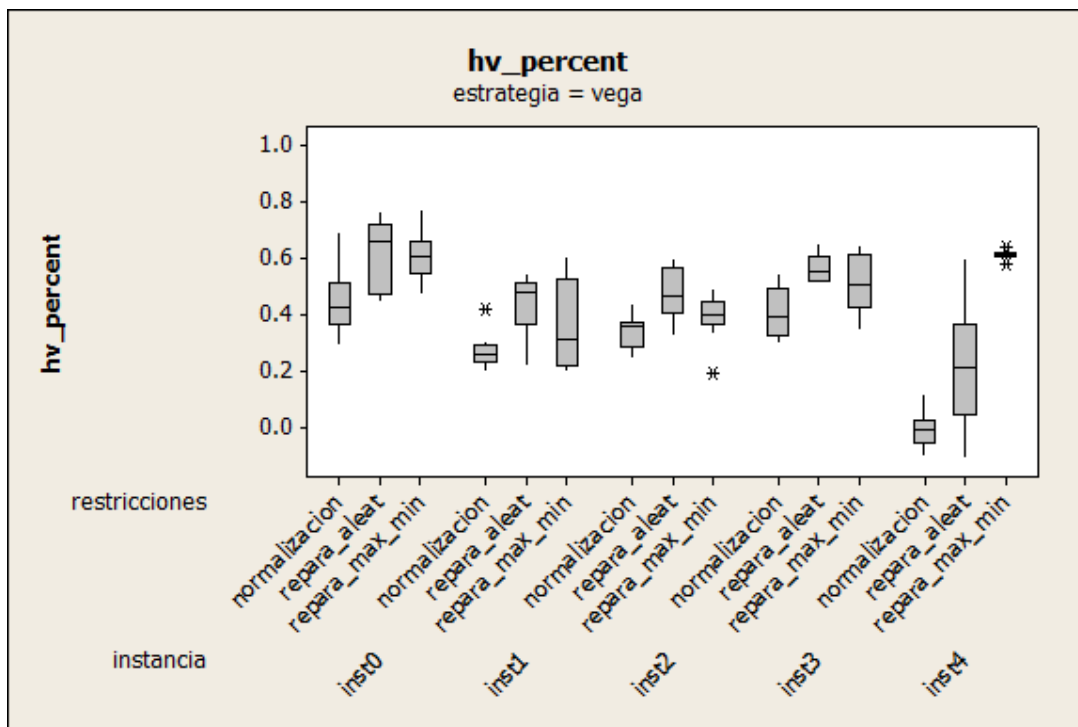


Figura 3.21 HV_{perc} manejo de restricciones (VEGA)



Lo anterior se resume en la Tabla 3.6 donde "x" significa que la técnica indicada no es la mejor, "✓" que es la mejor y "o" que el resultado es inconcluso. De este modo es que se ha optado por seleccionar el método de reparación aleatoria para las implementaciones de las metaheurísticas.

Tabla 3.6 Análisis cualitativo: selección de un método de manejo de restricciones

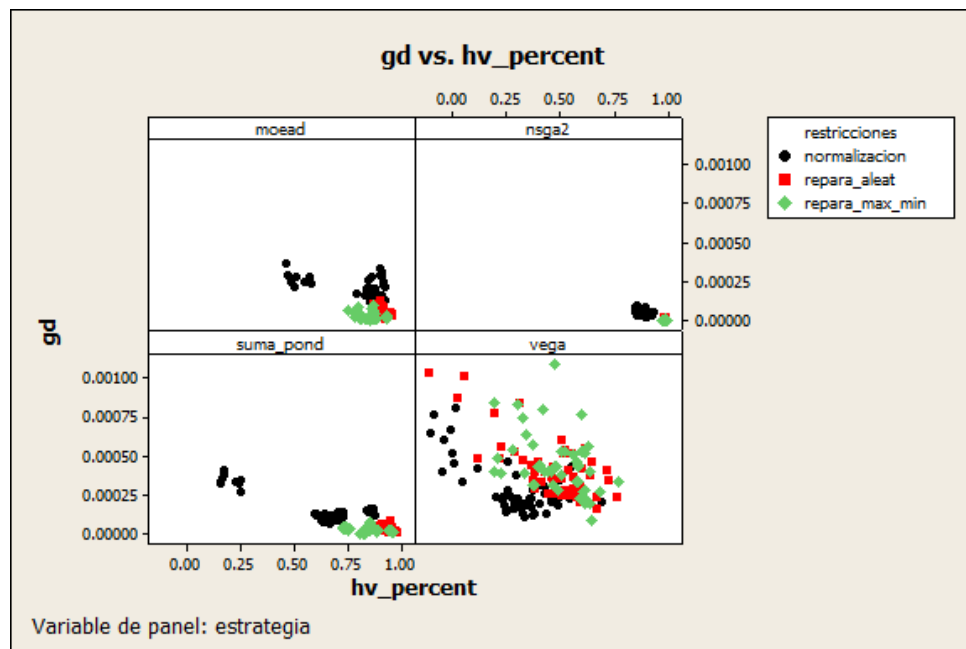
	normalización	repara_aleat	repara_max_min
VEGA	x	✓	x
Suma_pond	x	✓	x
MOEA/D	x	✓	x
NSGA2	x	o	o

Para entender la razón de que un método que involucra un proceso estocástico sea más eficiente, es necesario recordar que la reparación min-max impone quedarse con los activos de mayor peso en la cartera al reparar una solución. La intención es probar si este efecto sería positivo en el algoritmo, y de hecho lo fue, dado que obtuvo resultados significativamente mejores que la normalización. Por otro lado, reparar de forma estocástica no equivale a una simple generación de nuevas soluciones aleatorias, sino que se están seleccionando alelos que han presentado buenas características en la evolución de los algoritmos.

A continuación, en la Figura 3.22, Figura 3.23 y Figura 3.24 se presenta otra perspectiva gráfica de los resultados obtenidos. Considerando las tres métricas principales, se grafican los resultados en dos dimensiones, tomando en cuenta sucesivamente cada par de métricas posible, eso para cada técnica de manejo de restricciones y metaheurística. De este modo, los mejores resultados

tenderán a aproximarse a una esquina del gráfico, la cual se hará referencia como esquina de eficiencia, dependiendo si las métricas se maximizan o minimizan²³. De este modo, la agrupación de los puntos en un área menor refleja de una forma cualitativa, que la eficiencia de la estrategia es más estable, es decir, se logran resultados con una calidad muy similar al hacer múltiples ejecuciones.

Figura 3.22 Manejo de restricciones: gd vs. HV_{perc}



De acuerdo a lo anterior es posible hacer las siguientes conclusiones:

- Los resultados que obtiene la estrategia VEGA son muy variables, ya que los puntos están demasiado dispersos.
- Los métodos de manejo de restricciones reparación aleatoria y max-min tienen una cercanía muy similar a la esquina de eficiencia en la mayoría de

²³ Recuerde que ESS y gd se pretenden minimizar, mientras que HV_{perc} maximizar.

los gráficos, pero los de reparación aleatoria suelen ser más concentrados.

Una razón más por la cual se prefiera la modalidad aleatoria

- Finalmente NSGA-II es en el que se presenta una mayor densidad de los puntos en los gráficos (es más estable).

Figura 3.23 Manejo de restricciones: *gd* vs. *ESS*

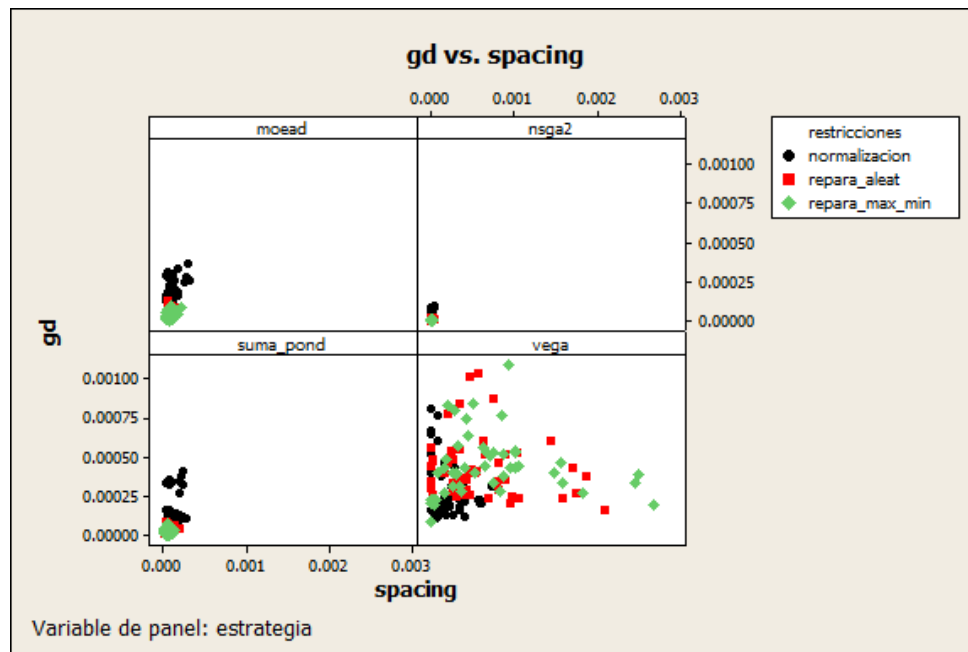
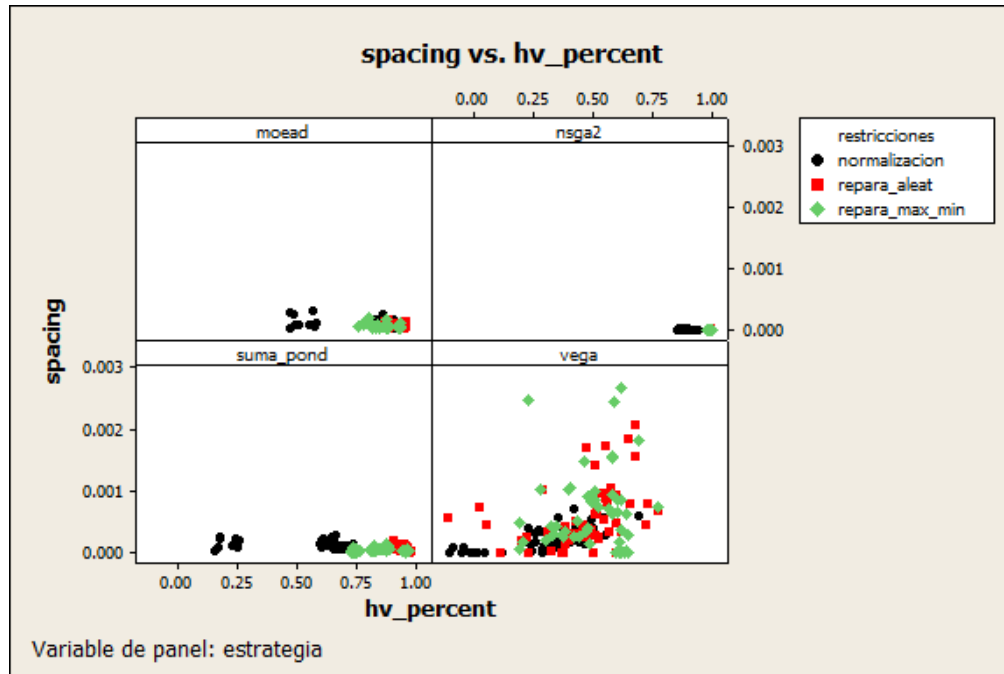


Figura 3.24 Manejo de restricciones: ESS vs. HV_{perc} 

3.7 Ajuste de parámetros de las técnicas para el problema

En este apartado se resumen los resultados obtenidos del análisis para el ajuste de parámetros, realizados para la técnica de Algoritmo Genético con Suma Ponderada de Objetivos (SPO). En el apéndice Ajuste de parámetros para las estrategias VEGA, NSGA-II y MOEA/D se encuentran la descripción de procedimientos similares, realizados para las otras tres técnicas. La calibración se hizo con *inst0* e *inst2*. Se presentan de igual manera todas las pruebas de hipótesis realizadas donde fue requerido²⁴, las cuales tienen un nivel de significancia estadística de $\alpha = 0.05$. El criterio de paro es el número de llamadas a la

²⁴ Si desea conocer más a detalle la manera como se interpreta la información obtenida en las pruebas realizadas consulte el apéndice: Interpretación de resultados en las pruebas de hipótesis

función objetivo (limitado a 1000N llamadas para todos los experimentos de esta sección).

Para el ajuste de parámetros el orden de relevancia de las métricas es el siguiente:

- I. Hipervolumen
- II. Distancia generacional
- III. Spacing
- IV. Cardinalidad

El hipervolumen es el de mayor prioridad dado que es una métrica de convergencia-dispersión y es la única acorde al concepto de dominancia de Pareto.

3.7.1 Suma Ponderada de Objetivos

Valores extremos

Para la primera etapa de ajuste se eligieron los valores extremos e intermedios de los parámetros como se muestra en la Tabla 3.7, tales valores son los mismos utilizados en la etapa1 de ajuste en todas las metaheurísticas.

Tabla 3.7 Valores de los parámetros etapa1

Parámetro	Valores
Tasa de mutación (zm)	{0.001, 0.05, 0.2, 0.5}
Tamaño de la población	{10, 100, 1000}

En la implementación de esta técnica se usaron 21 valores para el vector λ . La unidad de medida para el tiempo, en el resto del documento está dada en segundos (s).

Del criterio de paro ya se ha comentado que es el número de llamadas a la función objetivo, pero a partir de este punto empezará a ser ineludible tomar en cuenta el tiempo. Los resultados obtenidos en función del hipervolumen se muestran en la Figura 3.25, es difícil hacer una conclusión a grandes rasgos, pero de acuerdo a la Figura 3.26 sí hay una influencia del tamaño de la población en los tiempos de ejecución. Posteriormente será notado que este efecto no sucede con todas las estrategias. Por otro lado, de acuerdo a la Figura 3.27, no es relevante el aumento del tiempo en la calidad de la solución después de cierto tamaño de población, es decir, aun utilizando más tiempo, no hay una mejora en la calidad de las soluciones.

Figura 3.25 Ajuste parámetros etapa1: HV_{perc} (SPO) (a)

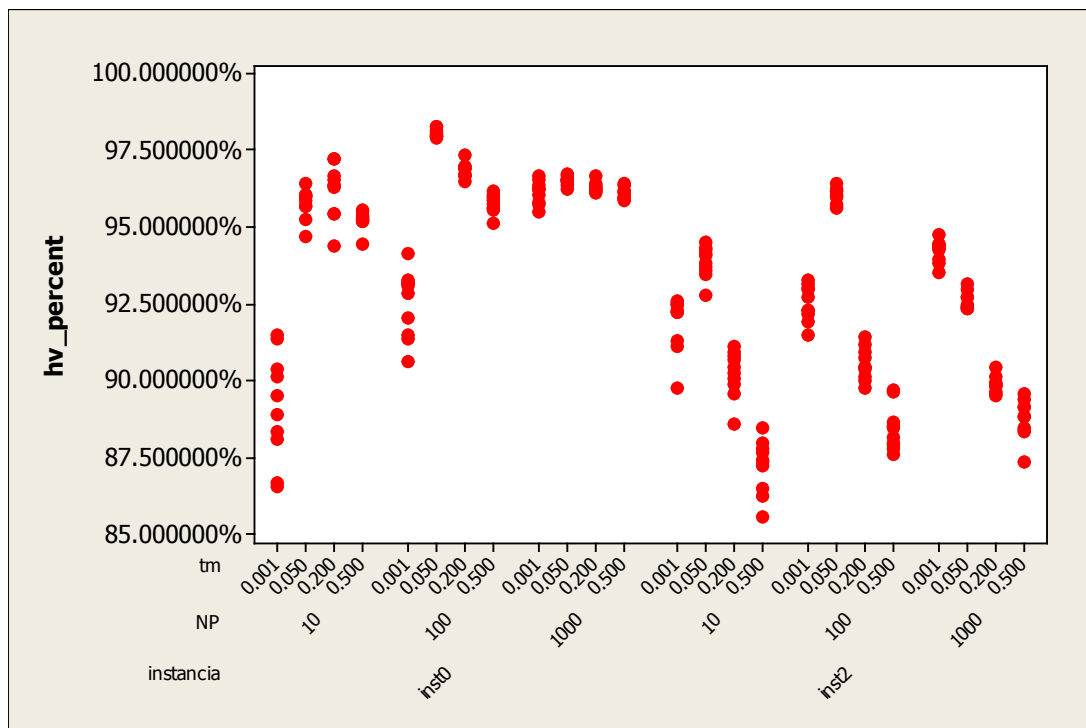


Figura 3.26 Ajuste parámetros etapa1: tiempo (SPO)

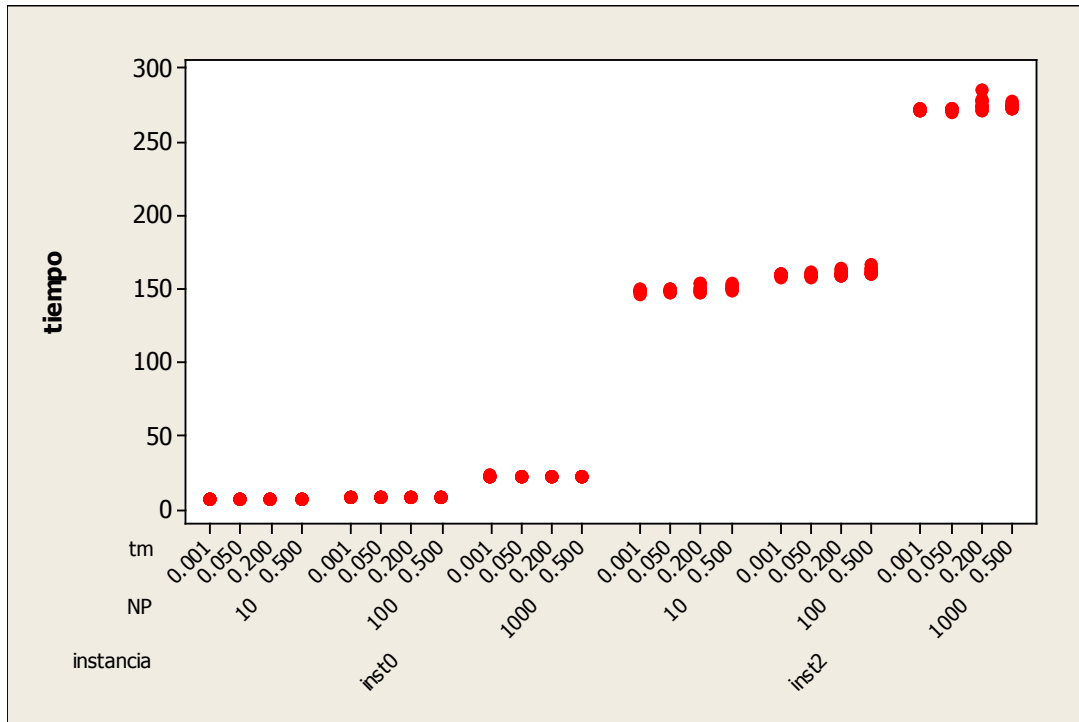
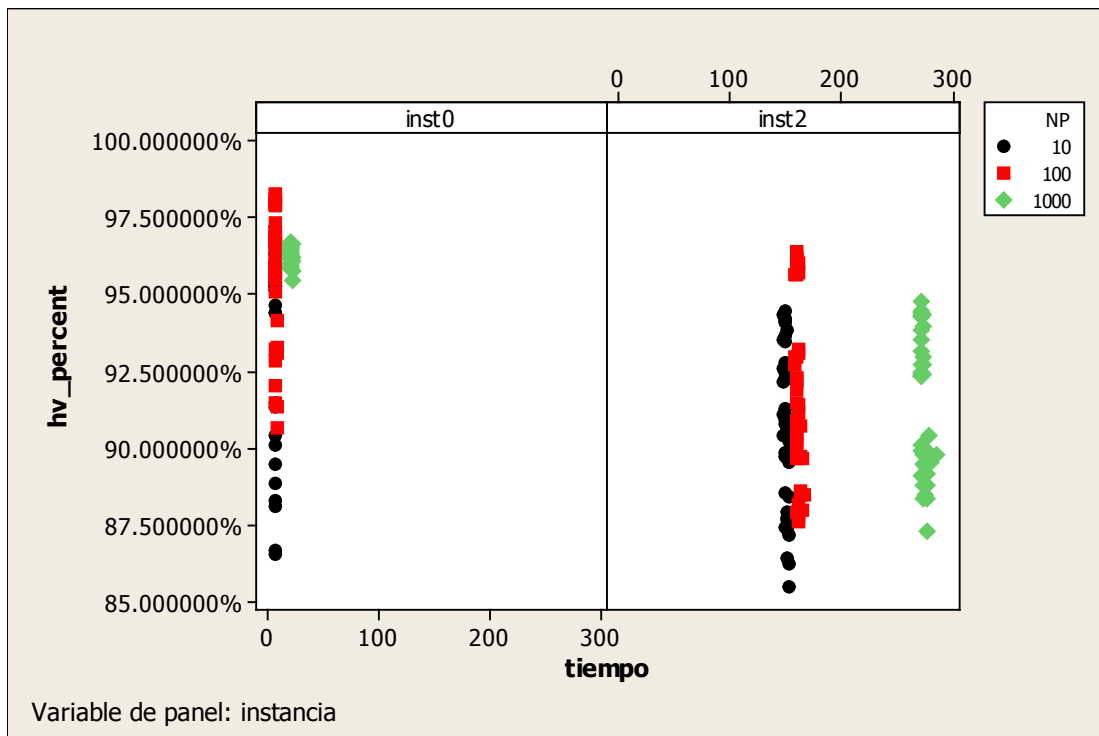


Figura 3.27 Ajuste parámetros etapa1: HV_{perc} vs. tiempo (SPO)



Efectivamente, observando también la tendencia del parámetro tamaño de la población en la Figura 3.27, el valor del hipervolumen parece aumentar de $NP = 10$ a $NP = 100$ y luego disminuir de $NP = 100$ a $NP = 1000$ alcanzando los más altos valores en 100, el cual de esta forma, es elegido en la presente etapa.

En esta técnica el parámetro zm no tiene un efecto significativo en el tiempo (Figura 3.26), de acuerdo a la anterior elección la información se reduce como se muestra en la Figura 3.28, es sencillo concluir que los mejores resultados son para $zm = 0.05$.

En la Figura 3.29 se presentan los resultados para las otras métricas. El mismo efecto de HV_{perc} se presenta en gd , mientras que ESS muestra mejores valores para tasas bajas y, por otro lado el aumento de zm reduce la cardinalidad²⁵.

²⁵ Recuerde que la cardinalidad alta no garantiza convergencia o dispersión.

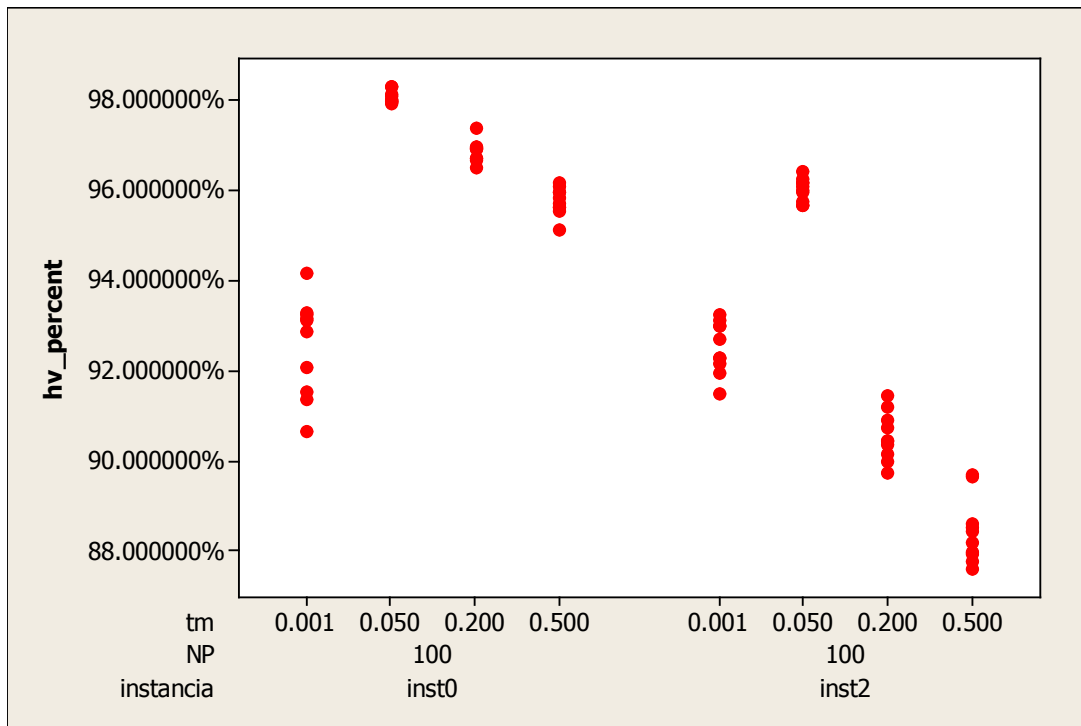
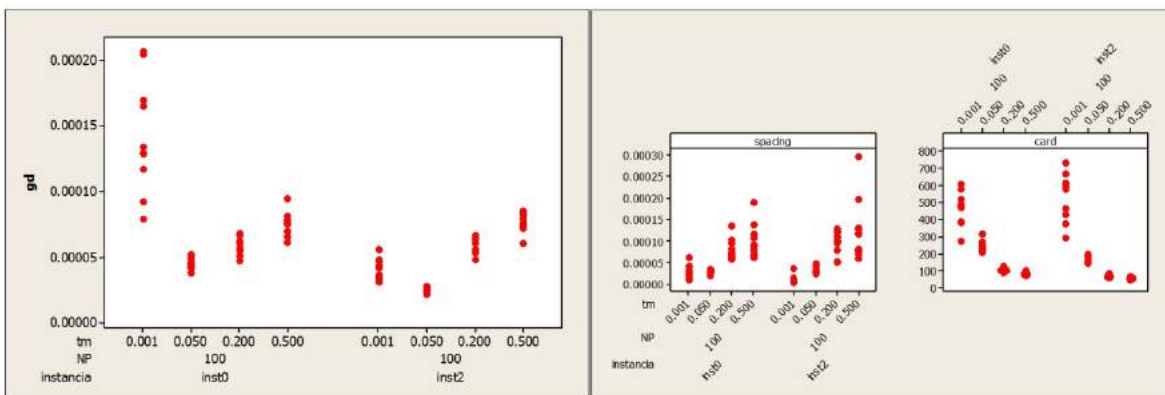
Figura 3.28 Ajuste parámetros etapa1: HV_{perc} (SPO) (b)

Figura 3.29 Ajuste parámetros etapa1: gd, ESS y card (SPO)



Diseño factorial completo

Para la segunda etapa de la actual técnica, los valores probados fueron los que aparecen en la Tabla 3.8. Los que están en negritas son los elegidos en la etapa anterior.

Tabla 3.8 Valores de los parámetros etapa2 (SPO)

Parámetro	Valores
Tasa de mutación (zm)	{0.01, 0.05 , 0.1}
Tamaño de la población	{20, 30, 100 , 500}

Al mirar la gráfica de tiempo (Figura 3.30), es notable que las pruebas muestren tiempos más parecidos, aunque sigue siendo cierto que una población más grande hace más costosa la ejecución. Sin embargo, observe la Figura 3.31, para el tamaño de población más grande las soluciones no muestran una calidad superior. Esto aunado a que en cuestión temporal es más costoso, se descarta el valor de 500 para tal parámetro.

Figura 3.30 Ajuste parámetros etapa2: tiempo (SPO)

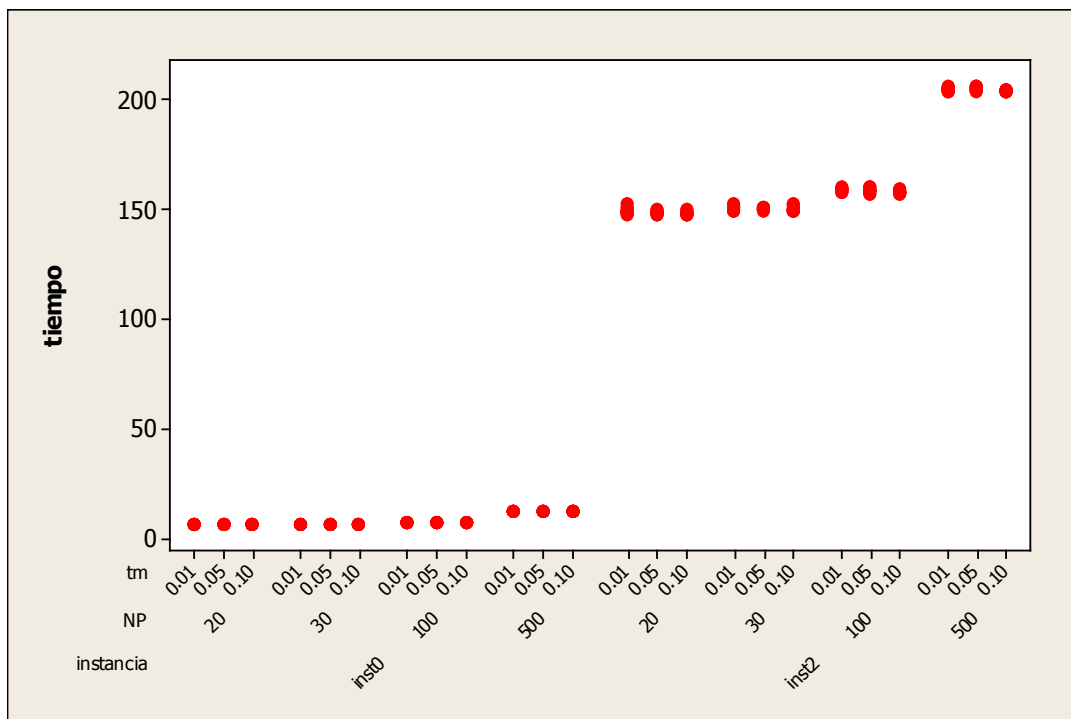
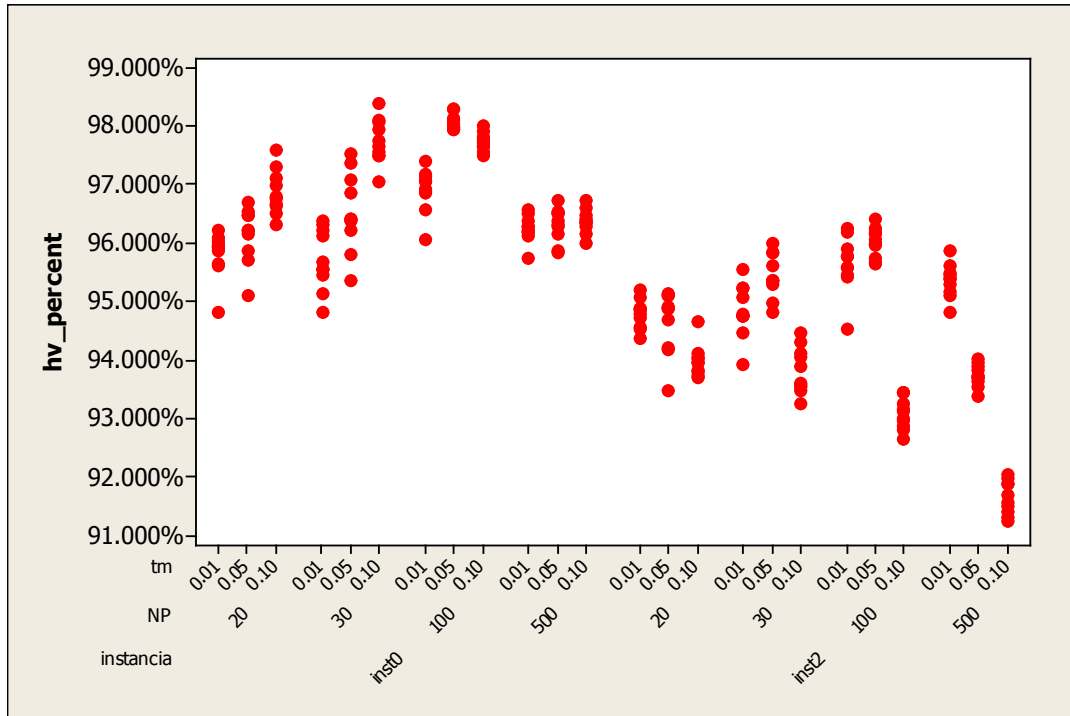


Figura 3.31 Ajuste parámetros etapa2: HV_{perc} (SPO) (a)

De acuerdo a la decisión anterior, la información ahora se puede ver más claramente en la Figura 3.32. Por ahora no es posible hacer una conclusión certera respecto a NP , la tendencia de la tasa de mutación no se muestra igual en todos los tamaños de población, pero en cada uno de ellos hay un valor que muestra mejor desempeño. De acuerdo a la clasificación propuesta en la Tabla 3.9 (donde 1 significa que el valor para zm fue el mejor en el conjunto de experimentos y 0 lo contrario), es elegida la tasa de mutación correspondiente a la mayor suma de las columnas, es decir $zm = 0.05$.

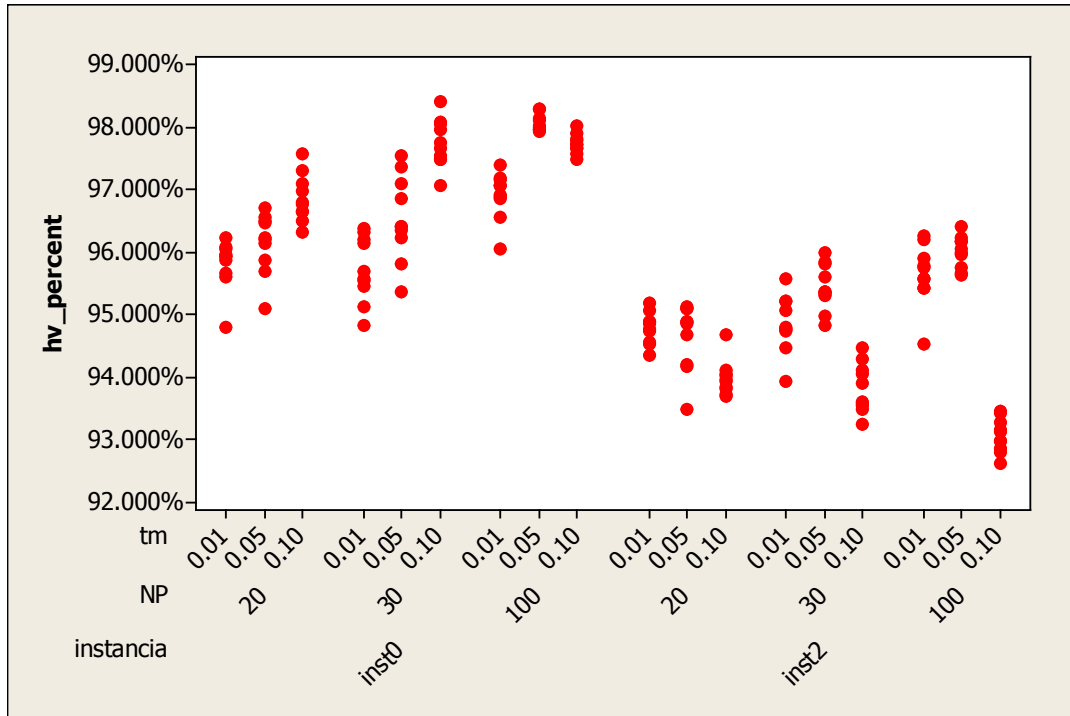
Figura 3.32 Ajuste parámetros etapa2: HV_{perc} (SPO) (b)

Tabla 3.9 Ajuste parámetros etapa2: selección no determinista de una tasa de mutación

Conjunto de experimentos		Tasas de mutación		
Instancia	Ψ	0.01	0.05	0.1
inst0	20	0	0	1
inst0	30	0	0	1
inst0	100	0	1	0
inst1	20	1	0	0
inst1	30	0	1	0
inst1	100	0	1	0
Total		1	3	2

Teniendo fijo el valor de zm únicamente queda por elegir a NP a partir de la información de la Figura 3.33, la tendencia es clara, para inst0 $NP = 100$, por otro lado para inst2 se hizo la prueba de la Tabla 3.10, donde es posible rechazar la hipótesis $H_0: HV_{perc}(NP = 30) \geq HV_{perc}(NP = 100)$ con un nivel de significancia

de 0.05²⁶ (para mayores detalles sobre cómo se realizaron las pruebas de hipótesis en la presente investigación se refiere el lector al apéndice Interpretación de resultados en las pruebas de hipótesis).

Figura 3.33 Ajuste parámetros etapa2: HV_{perc} (SPO) (c)

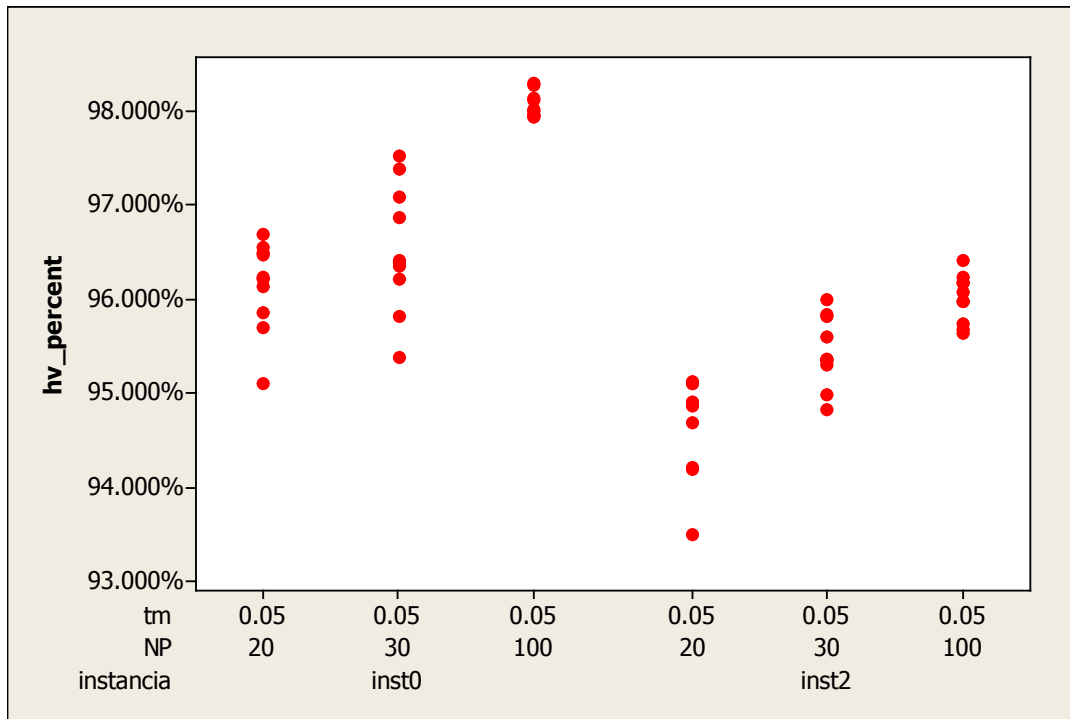


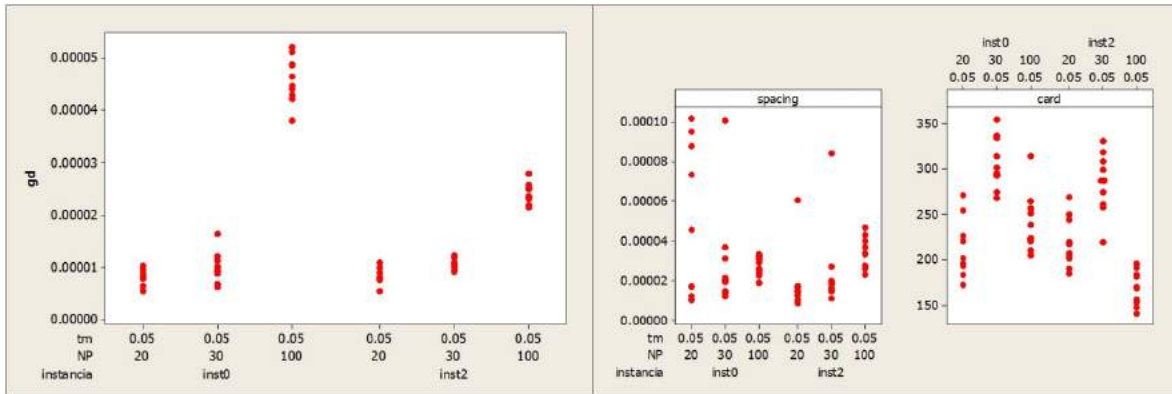
Tabla 3.10 Ajuste parámetros etapa2: Prueba de rangos de signos de Wilcoxon (SPO)

Two-sample wilcoxon rank-sum (Mann-whitney) test			
np	obs	rank sum	expected
30	10	66	105
100	10	144	105
combined	20	210	210
unadjusted variance	175.00		
adjustment for ties	0.00		
adjusted variance	175.00		
Ho: hv_per~t(np==30) = hv_per~t(np==100)			
z = -2.948			
Prob > z = 0.0032			

²⁶ Con $z < 0$ y $\frac{p}{2} < 0.05$ se puede realizar la prueba de cola izquierda y rechazar la hipótesis nula H_0 contra la hipótesis alterna $H_1 HV_{perc}(30) < HV_{perc}(100)$

En la Figura 3.34 se muestran las métricas restantes para los valores de los parámetros seleccionados, observe que gd muestra una tendencia de deterioro con el aumento de NP , aunque no sucede lo mismo con la cardinalidad. Además, cabe mencionar que las variaciones en gd son minúsculas, esto debido a que la convergencia ya es muy alta para todas las pruebas: en la etapa anterior se llegaban a valores de hasta 0.0002, mientras que en la presente ninguno es mayor que 0.00006.

Figura 3.34 Ajuste parámetros etapa2: gd , ESS y card (SPO)



Para las técnicas restantes se realizó la misma metodología de las dos etapas, la diferencia fue que, en el análisis comparativo, los criterios no se prestaron a ser manejados de la misma manera; esto debido a las particularidades del modo operativo de cada una de las metaheurísticas, que las vuelven sensibles a la variación de los parámetros de diferentes maneras. De igual modo, en la etapa de calibración de parámetros, se logró obtener diferentes conclusiones acerca de las particularidades mencionadas. Para revisar con mayor detalle la metodología de las técnicas restantes consulte el apéndice: Ajuste de parámetros para las estrategias VEGA, NSGA-II y MOEA/D.

De acuerdo al proceso de ajuste de parámetros se pueden resumir los resultados como se muestra en la Tabla 3.11. Los obtenidos en la Etapa2 fueron usados para las ejecuciones en la fase experimental de la presente investigación.

Tabla 3.11 Valores finales de los parámetros

Estrategia	Parámetro	Etapas1	Etapas2
SPO	zm	0.05	0.05
SPO	NP	100	100
VEGA	zm	0.05	0.01
VEGA	NP	1000	1000
NSGA-II	zm	0.05	0.1
NSGA-II	NP	100	30
MOEA/D	zm	0.2	0.1
MOEA/D	NP	100	100
MOEA/D	θ	1000	10
MOEA/D	T		10
MOEA/D	T_{perc}		10.00%

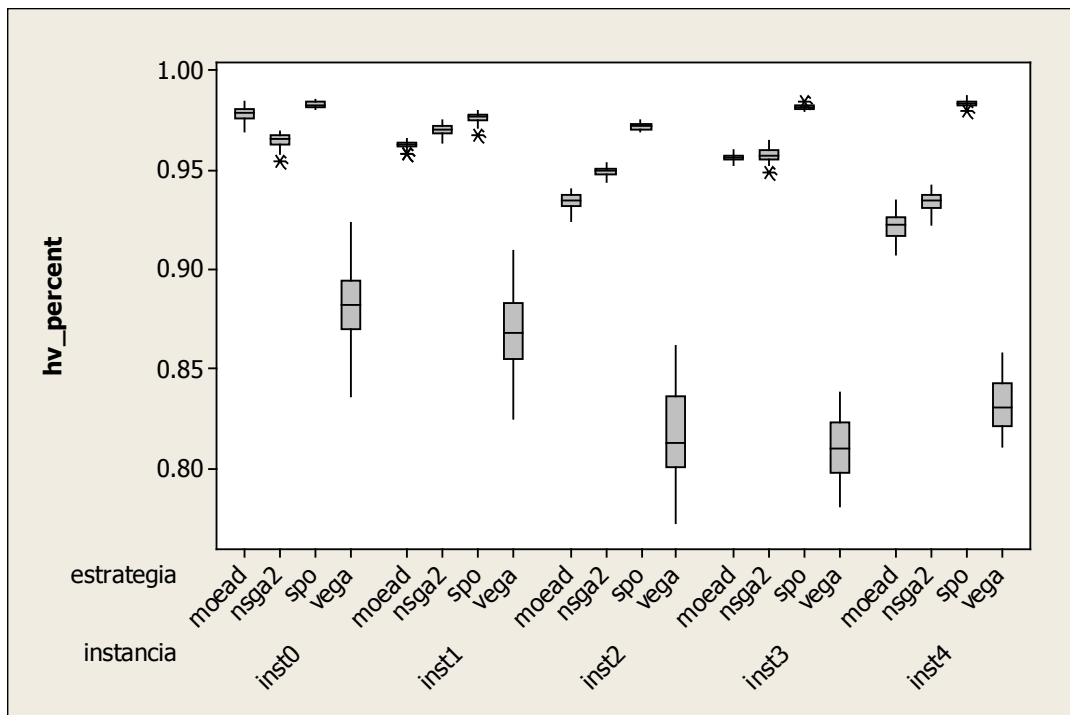
4 Desarrollo experimental

En el presente capítulo se desarrolla la etapa experimental final y el análisis de eficiencia para cada una de las técnicas probadas sobre las instancias del problema del portafolio de inversión presentadas anteriormente.

4.1 Implementación 1: Ajuste temporal

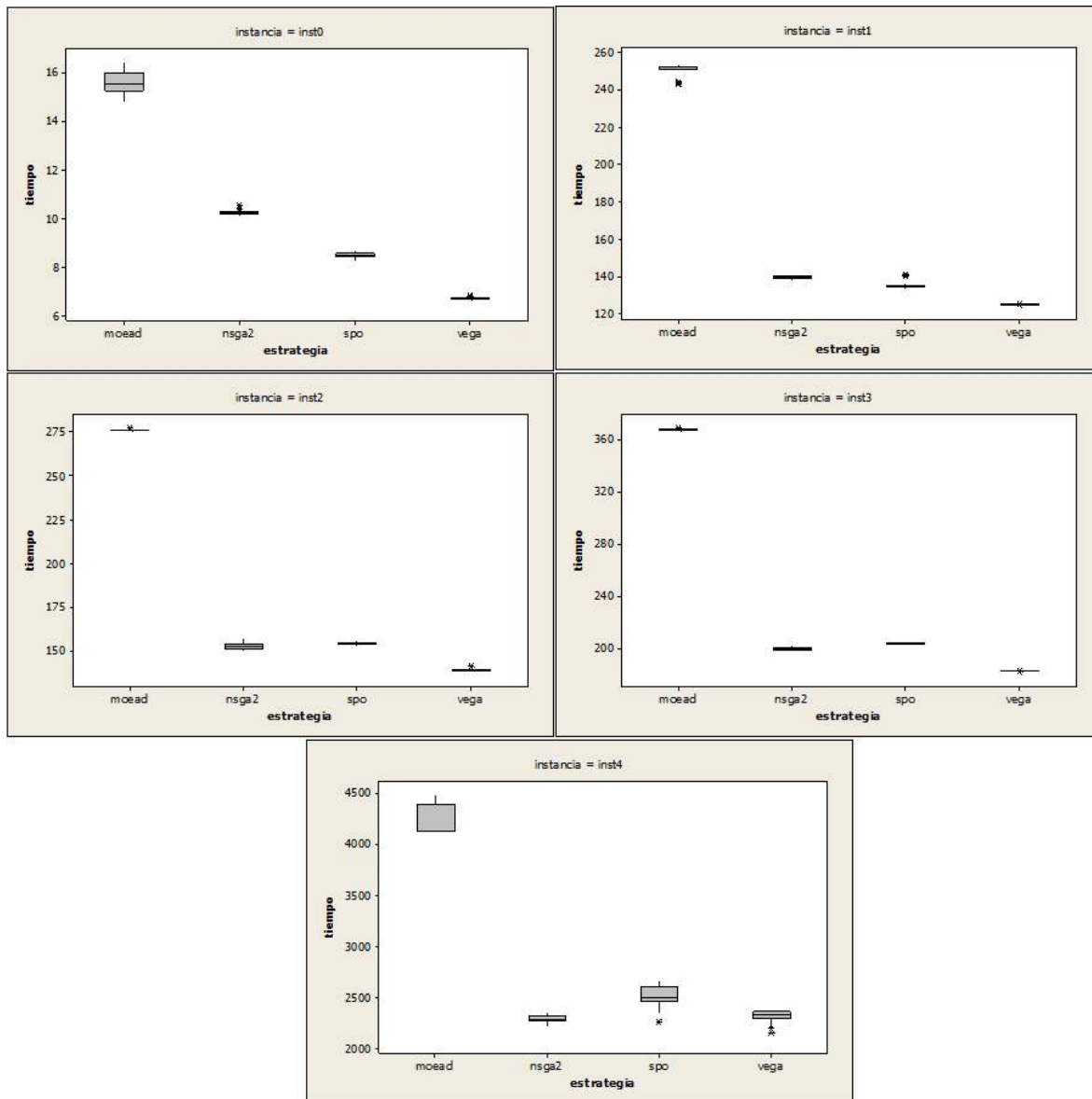
Para cada instancia resuelta se realizaron 30 experimentos por técnica metaheurística, de acuerdo a las características y valores para los parámetros elegidos en el anterior capítulo. Para el parámetro HV_{perc} se muestran los resultados en la Figura 4.1.

Figura 4.1 Implementación 1: HV_{perc}



Se mencionó en el capítulo anterior que definir un criterio de paro como el número de llamadas a la función objetivo se justifica por el requisito de ejecutar a los algoritmos en las condiciones lo más similares posibles en cuanto a recursos computacionales utilizados. De tal forma, se puede realizar una comparación de desempeño justa. Sin embargo, en las primeras ejecuciones del presente capítulo, de acuerdo a la Figura 4.2, es visible que en cuestión temporal las condiciones no son del todo similares, ya que MOEA/D utiliza una mayor cantidad de tiempo de ejecución, mientras que VEGA mucho menos. Pareció preciso, al igual que en la etapa de ajuste de parámetros, hacer un intento por igualar los tiempos de ejecución, sin perder el número de llamadas a la función objetivo como el criterio de mayor relevancia.

Figura 4.2 Implementación 1: tiempo



Al observar los estadísticos temporales resumidos en la Tabla 4.1, se puede notar, a grandes rasgos, bastante similitud entre la media y mediana. Esta tendencia se confirma en la prueba de hipótesis presentada en la Tabla 4.2, en la cual no es posible rechazar la hipótesis de igualdad entre media y mediana, por lo que no es señalada una diferencia estadísticamente significativa entre las dos medidas centrales para la variable temporal.

Tabla 4.1 Implementación 1: estadísticos del tiempo (s)

estrategia	instancia	media	desv. Est.	mediana
MOEA/D	inst0	15.5802	0.4395	15.5256
MOEA/D	inst1	250.5194	3.1921	251.5567
MOEA/D	inst2	276.3875	0.2559	276.3847
MOEA/D	inst3	367.8605	0.2946	367.7866
MOEA/D	inst4	4239.7671	140.0731	4128.6927
NSGA-II	inst0	10.2569	0.0755	10.2456
NSGA-II	inst1	139.5313	0.7144	139.5235
NSGA-II	inst2	152.7760	2.0082	152.6625
NSGA-II	inst3	199.5520	1.1006	199.3196
NSGA-II	inst4	2287.4725	37.0931	2291.2116
SPO	inst0	8.4907	0.0944	8.4838
SPO	inst1	135.0027	1.6340	134.5455
SPO	inst2	154.5674	0.6611	154.4038
SPO	inst3	203.7893	0.5945	203.7001
SPO	inst4	2507.9305	96.7830	2503.9120
VEGA	inst0	6.7193	0.0321	6.7152
VEGA	inst1	124.8962	0.1162	124.8802
VEGA	inst2	139.2145	0.6926	139.4778
VEGA	inst3	182.2776	0.1578	182.2430
VEGA	inst4	2307.7614	62.2145	2331.1319

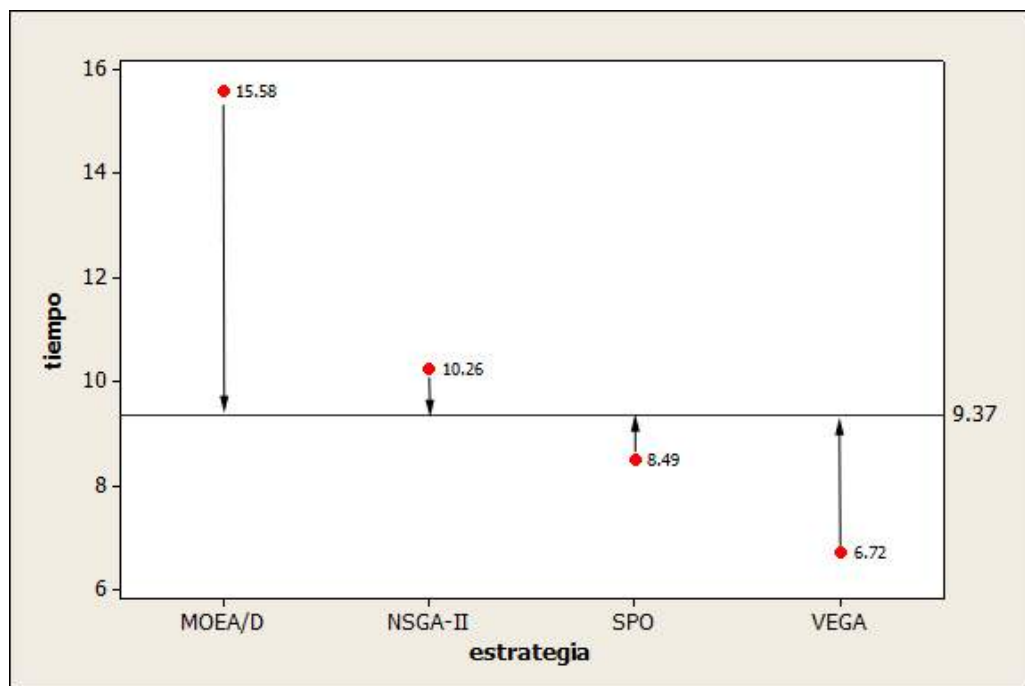
Dicho lo anterior, la elección entre las dos medidas centrales no es significativamente trascendental, por lo que se propone elegir la media. Si ahora se presta atención a los resultados en negritas de la Tabla 4.1, las estrategias NSGA-II y SPO, como ya se mencionó en el capítulo anterior, muestran, en promedio, tiempos similares en cada instancia, mientras que para las restantes no es así. A partir de tales valores (NSGA-II y SPO), se pretende hacer un reajuste en el tiempo de las otras dos estrategias y así lograr un consumo de recursos computacionales similar para las cuatro técnicas.

Tabla 4.2 Implementación 1: prueba Wilcoxon tiempo

Two-sample wilcoxon rank-sum (Mann-whitney) test			
medida	obs	rank sum	expected
media	20	416	410
mediana	20	404	410
combined	40	820	820
unadjusted variance	1366.67		
adjustment for ties	0.00		
adjusted variance	1366.67		
Ho: valor(medida==media) = valor(medida==mediana)			
z =	0.162		
Prob > z =	0.8711		

La metodología para hacer el ajuste se ilustra en el ejemplo de la Figura 4.3. La gráfica muestra los promedios del tiempo de ejecución obtenidos en el actual apartado para inst0. La intención es buscar un punto medio entre el tiempo de ejecución del NSGA-II y SPO $\left(\frac{10.2569+8.4907}{2} = 9.3738\right)$ y hacer que las otras dos estrategias se ajusten a este valor como se muestra en la Figura 4.3

Figura 4.3 Ejemplo ajuste temporal inst0 (segundos)

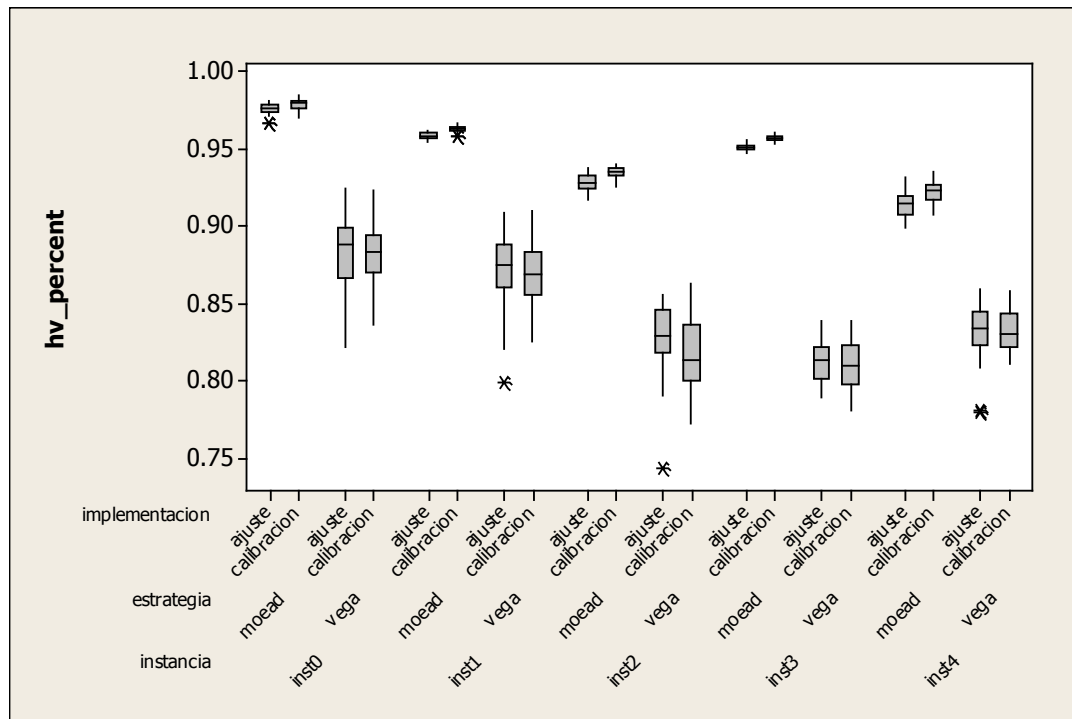


Los promedios para el ajuste temporal (valores en negritas de la Tabla 4.1) se muestran nuevamente en la Tabla 4.3. A partir de ellos, se calcula el punto intermedio en la cuarta columna, y finalmente estos valores serán el tiempo de ejecución estimado para las estrategias MOEA/D y VEGA.

Tabla 4.3 Ajuste de tiempo para MOEA/D y VEGA

instancia	NSGA-II	SPO	promedio
inst0	10.2569	8.4907	9.3738
inst1	139.5313	135.0027	137.2670
inst2	152.7760	154.5674	153.6717
inst3	199.5520	203.7893	201.6706
inst4	2287.4725	2507.9305	2397.7015

En la nueva implementación con los tiempos ajustados, la calidad de los resultados obtenidos bajo la métrica HV_{perc} se encuentra resumida en la Figura 4.4, donde el grupo llamado *calibración* corresponde a las 1000N llamadas a la función objetivo para todas las instancias, mientras que el grupo *ajuste* a la corrección de los tiempos de acuerdo a la Tabla 4.3.

Figura 4.4 Implementación 1: HV_{perc} comparación con ajuste temporal

Es posible notar que la diferencia entre las dos estrategias, con y sin tiempo corregido, es muy poca en cuanto a calidad de la soluciones para el MOEA/D, a pesar de restringir el tiempo hasta un 45% menos (Tabla 4.4). En cuanto a VEGA, sucede algo similar pero de forma opuesta, la implementación con tiempo corregido no es significativamente mejor, aunque en esta técnica no se aumenta el tiempo de ejecución a más del 11% en cuatro de cinco instancias de acuerdo a la Tabla 4.4.

Tabla 4.4 Magnitud de la variación al corregir el tiempo

instancia	1000N llamadas a la función objetivo (tiempo promedio)		tiempo corregido	variación de tiempo	
	MOEA/D	VEGA		MOEA/D	VEGA
inst0	15.5802	6.7193	9.3738	-39.84%	39.50%
inst1	250.5194	124.8962	137.267	-45.21%	9.90%
inst2	276.3875	139.2145	153.6717	-44.40%	10.38%
inst3	367.8605	182.2776	201.6706	-45.18%	10.64%
inst4	4239.7671	2307.7614	2397.7015	-43.45%	3.90%

VEGA tiende a tener marcadas variaciones en la aceleración de sus métricas con el paso de las generaciones²⁷ (revisar el concepto de aceleración en el apéndice El problema del tamaño de archivo EP), por eso es que, al permitir más tiempo de ejecución, se logra alcanzar mejores resultados, pero con una mayor variabilidad²⁸ (nótese que las métricas de los resultados de VEGA siempre tienen una dispersión muy fuerte).

4.2 Implementación 2: resultados finales

Las implementaciones con tiempo corregido son las presentadas como resultados finales en el presente apartado, debido a que cumplen con las características y estándares propuestos y justificados en los análisis previos, tanto de calibración, como de criterio de paro.

Recuerde que la ventaja de manejar el hipervolumen en porcentaje es que no se basa en unidades. De este modo, bajo el supuesto de que la disponibilidad de recursos es equitativa, es posible hacer comparativos entre

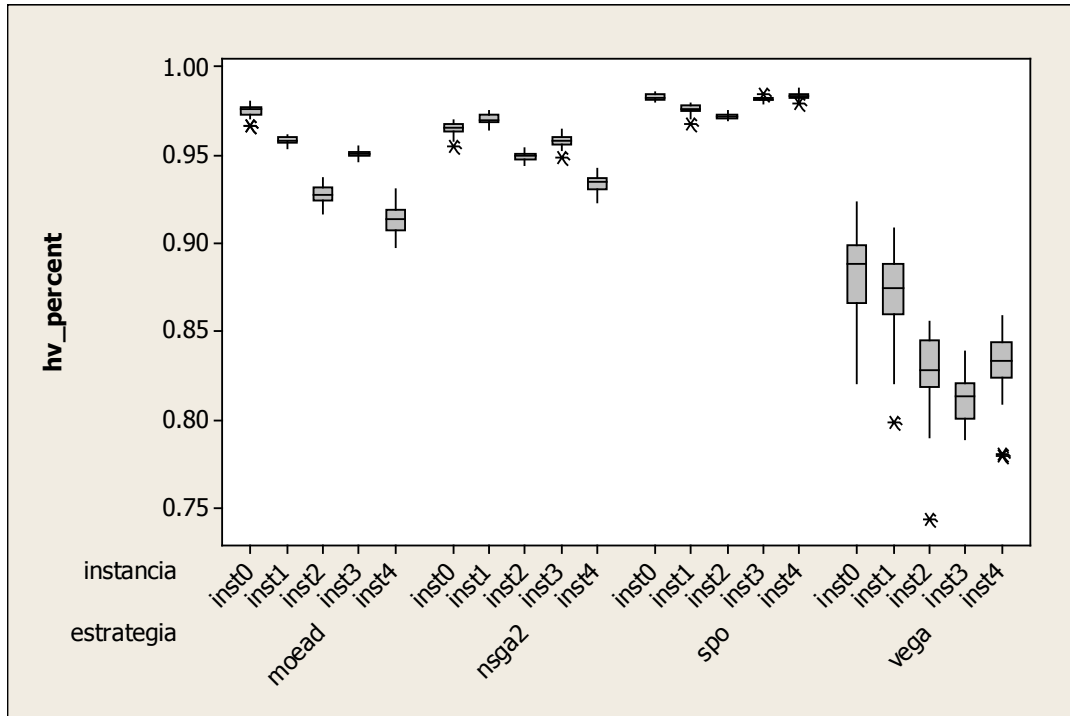
²⁷ Al estar cambiando la orientación del proceso de optimización de un objetivo a otro dado que cada subpoblación se especializa en un objetivo, en cada nueva generación hay variaciones bruscas en la calidad del frente aproximado, como puede mejorar mucho de un tajo, como puede suceder lo contrario.

²⁸ Cabe aclarar que las poblaciones iniciales son las mismas

instancias y determinar si una es más difícil que otra comparando la magnitud de tal métrica en porcentaje²⁹. Bajo ese argumento, es notable cómo cada estrategia se comporta de forma diferente de acuerdo a las características de la instancia: en la Figura 4.5 (solo se presenta la implementación con tiempos ajustados) se muestra la métrica HV_{perc} . Para MOEA/D y NSGA-II, inst4 fue la de mayor dificultad; para SPO lo fue inst2 mientras que a VEGA le corresponde la menor calidad de resultados en la inst3. Aunque el tamaño de la instancia es crucial en la dificultad del problema, otras características del frente de Pareto (ver Figura 1.2) también tendrán relevancia en la eficacia con la que particularmente cada estrategia resuelva el problema, y ese es un factor que no hay que perder de vista al elegir una técnica para resolver un MOP³⁰. Observe de igual manera, que con respecto a HV_{perc} ya se puede notar diferencias significativas en cuanto a calidad de los resultados.

²⁹ En otras métricas no es posible asegurar eso dado que una medición en una instancia específica puede no tener el mismo significado en calidad para otra instancia (e.g. $gd = 0.001$ puede significar una buena convergencia para una instancia, mientras que para otra podría significarse un resultado terrible)

³⁰ Tal efecto en la dificultad de la instancias también es notable en la implementación 1, donde a todas las ejecuciones les correspondían 1000*N* llamadas a la función objetivo (ver Figura 4.1)

Figura 4.5 Implementación 2: HV_{perc} 

Las métricas gd y ESS pueden ser comparadas entre cada una de las estrategias pero solo si se hace referencia a la misma instancia, en la Figura 4.6, Figura 4.7 y Figura 4.8 se muestran las métricas restantes.

De acuerdo a la métrica distancia generacional (Figura 4.6), nuevamente se observan los mejores resultados con SPO y los peores con VEGA. Por otro lado ESS en la Figura 4.7 suele variar con mayor dispersión aunque refleja conclusiones similares. Finalmente la particularidad de la métrica cardinalidad (Figura 4.8) es que SPO muestra resultados con muchos puntos, por otro lado, en NSGA-II el tamaño de la población es un factor sumamente crucial, por ende siempre se mantiene pequeña.

Figura 4.6 Implementación 2: Distancia Generacional

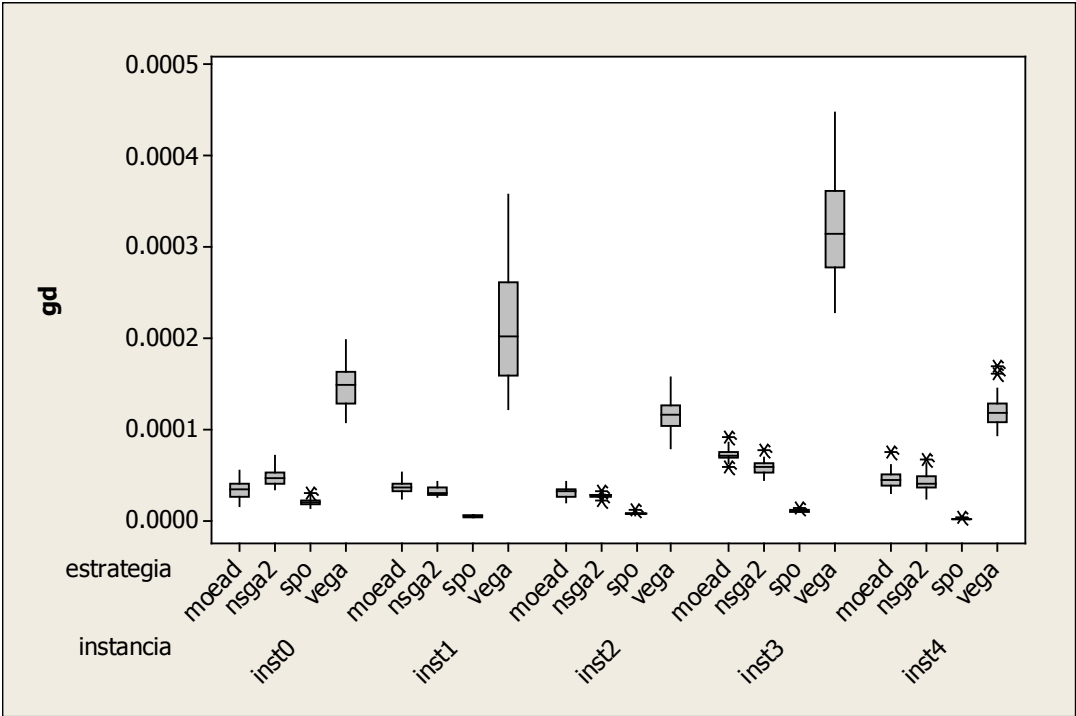


Figura 4.7 Implementación 2: ESS

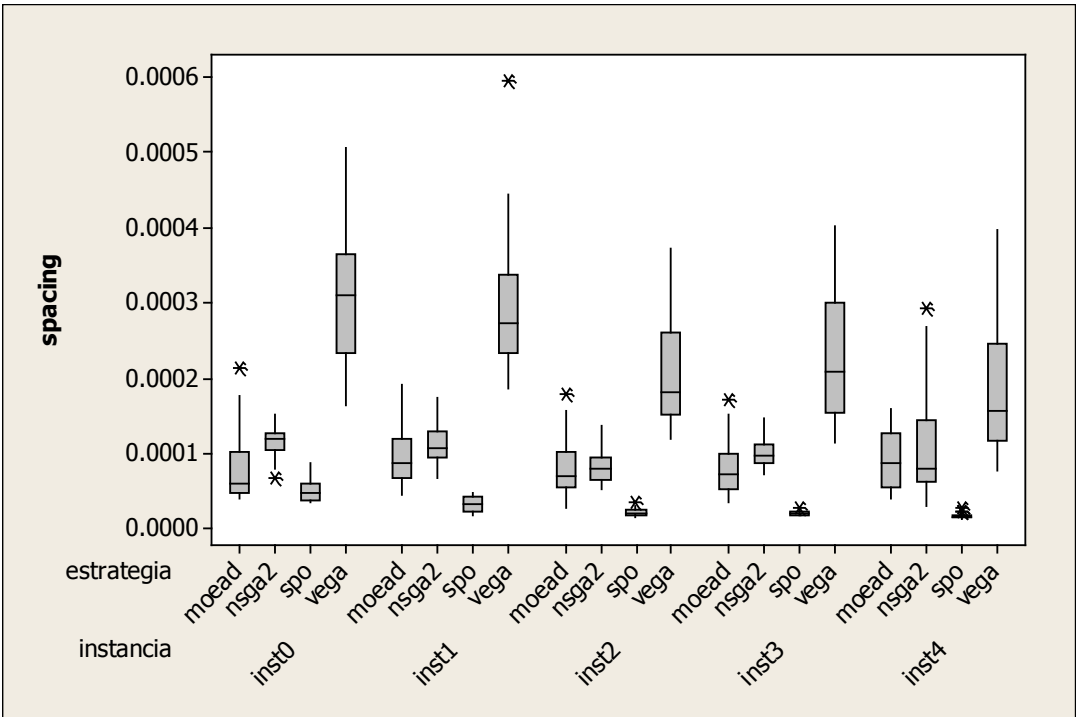
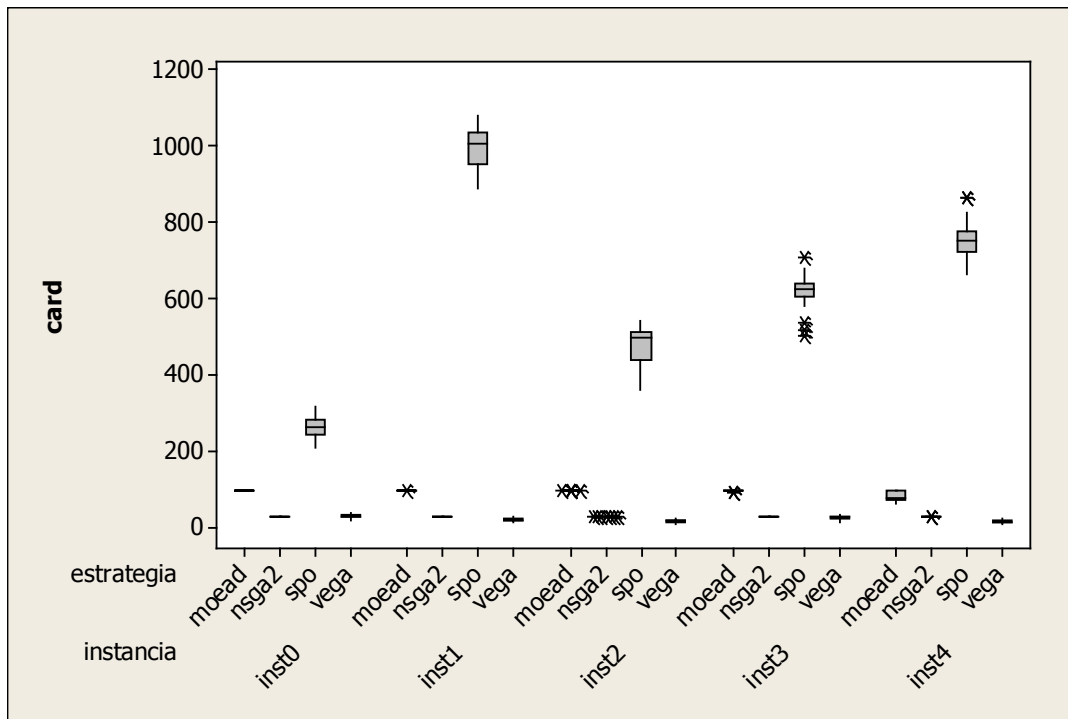


Figura 4.8 Implementación 2: Cardinalidad



4.3 Análisis de decisión bajo el criterio Hipervolumen

El objetivo del presente apartado es presentar un análisis comparativo de los resultados finales obtenidos en la sección 4.2. Se ha decidido trabajar desde diferentes enfoques, a continuación se usa el Hipervolumen como criterio de desempeño.

4.3.1 Eficiencia de las Metaheurísticas

De acuerdo a una medida central de la métrica (media y mediana), se decide el orden de desempeño de los algoritmos probados. La prueba de hipótesis en la Tabla 4.5, indica que no se puede rechazar la hipótesis de igualdad entre los valores de las dos medidas centrales, por lo tanto se concluye que es irrelevante cuál de las dos sea elegida como criterio. Se ha decidido por la

mediana, debido a que es la que se reporta en los gráficos de caja, los cuales son los de mayor uso en el presente documento. Además, corresponde a un elemento en específico de la muestra (o el promedio de dos si el tamaño de la muestra es par).

Tabla 4.5 Prueba de Wilcoxon Media vs Mediana (HV_{perc})

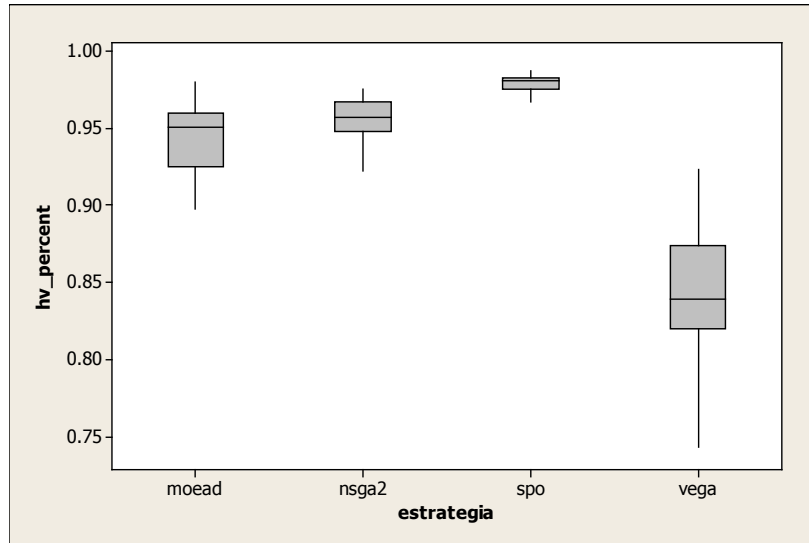
Two-sample wilcoxon rank-sum (Mann-whitney) test			
medida	obs	rank sum	expected
media	20	407	410
mediana	20	413	410
combined	40	820	820
unadjusted variance	1366.67		
adjustment for ties	0.00		
adjusted variance	1366.67		
Ho: valor(medida==media) = valor(medida==mediana)			
z = -0.081			
Prob > z = 0.9353			

Dicho lo anterior se sintetizan los resultados en la Tabla 4.6. En la columna “posición”, se muestra un ranking de eficiencia de las estrategias en cada una de las instancias, es decir, 1 le corresponde a la mejor y 4 a la peor técnica para la instancia respectiva. De acuerdo a esto, la mejor estrategia con respecto al hipervolumen es SPO, la peor es VEGA y MOEA/D únicamente logra ser superior a NSGA-II en inst0.

Tabla 4.6 Orden de desempeño de las técnicas (HV_{perc})

instancia	estrategia	media	desv.est.	coef. Var.	mediana	Mínimo	máximo	Posición
inst0	spo	98.31%	0.14%	0.14%	98.29%	98.07%	98.56%	1
inst0	moead	97.53%	0.35%	0.36%	97.60%	96.63%	98.05%	2
inst0	nsga2	96.50%	0.35%	0.36%	96.54%	95.49%	96.98%	3
inst0	vega	88.15%	2.41%	2.73%	88.81%	82.10%	92.38%	4
inst1	spo	97.65%	0.24%	0.24%	97.67%	96.78%	97.94%	1
inst1	nsga2	97.05%	0.28%	0.29%	97.04%	96.42%	97.56%	2
inst1	moead	95.84%	0.21%	0.21%	95.84%	95.38%	96.20%	3
inst1	vega	87.03%	2.48%	2.85%	87.51%	79.83%	90.84%	4
inst2	spo	97.22%	0.14%	0.14%	97.22%	96.99%	97.52%	1
inst2	nsga2	94.98%	0.23%	0.25%	94.96%	94.42%	95.39%	2
inst2	moead	92.78%	0.48%	0.51%	92.76%	91.74%	93.70%	3
inst2	vega	82.69%	2.31%	2.80%	82.86%	74.32%	85.53%	4
inst3	spo	98.18%	0.10%	0.10%	98.17%	97.98%	98.45%	1
inst3	nsga2	95.79%	0.33%	0.35%	95.78%	94.90%	96.48%	2
inst3	moead	95.08%	0.19%	0.20%	95.09%	94.70%	95.51%	3
inst3	vega	81.25%	1.32%	1.62%	81.35%	78.92%	83.84%	4
inst4	spo	98.39%	0.17%	0.17%	98.41%	97.99%	98.75%	1
inst4	nsga2	93.41%	0.47%	0.50%	93.47%	92.30%	94.23%	2
inst4	moead	91.37%	0.77%	0.84%	91.39%	89.82%	93.11%	3
inst4	vega	83.15%	1.95%	2.35%	83.34%	77.90%	85.90%	4

La métrica HV_{perc} permite normalizar el hipervolumen y comparar directamente los resultados obtenidos sobre instancias diferentes. Debido a esto, en la Figura 4.9 se presentan los resultados pero agrupados únicamente por técnica (todas las instancias están incluidas en cada técnica). De acuerdo a esta modalidad de comparación, se confirma que SPO sigue siendo la mejor técnica y la peor VEGA.

Figura 4.9 Desempeño de las técnicas (HV_{perc})

Entre NSGA-II y MOEA/D la prueba de la Tabla 4.7 indica que se rechaza la hipótesis de cola izquierda $H_0: HV_{perc}(MOEAD) \geq HV_{perc}(NSGA2)$ ya que $z = -3.811 < 0$ y $\frac{p}{2} = \frac{0.00001}{2} < \alpha = 0.05$ donde α es el nivel de significancia de la prueba³¹. En base a lo anterior, se concluye que NSGA-II tiene un desempeño significativamente superior al de MOEA/D.

Tabla 4.7 Prueba Wilcoxon MOEA/D vs NSGA-II (HV_{perc})

Two-sample wilcoxon rank-sum (Mann-whitney) test			
estrategia	obs	rank sum	expected
moead	150	19712	22575
nsga2	150	25438	22575
combined	300	45150	45150
unadjusted variance	564375.00		
adjustment for ties	0.00		
adjusted variance	564375.00		
Ho: hv_per~t(estrat~a==moead) = hv_per~t(estrat~a==nsga2)			
z = -3.811			
Prob > z = 0.0001			

³¹ Para mayores detalles ver apéndice: Interpretación de resultados en las pruebas de hipótesis

4.3.2 Análisis gráfico de los frentes de Pareto aproximados

Cada experimento realizado genera una aproximación del frente de Pareto de una instancia en particular, por lo cual no es posible presentarlos todos en el presente documento de forma gráfica. De modo que se enseñarán los frentes graficados correspondientes a la corrida mediana de cada estrategia en cada instancia de acuerdo a su desempeño en *HV*. Los frentes se muestran en el mismo gráfico dado que se aprecia de esta forma la cercanía con el frente real y entre estrategias, y también por separado, pues de este modo se observan mejor los rasgos en la forma del frente aproximado.

Figura 4.10 Inst0 mediana (a)

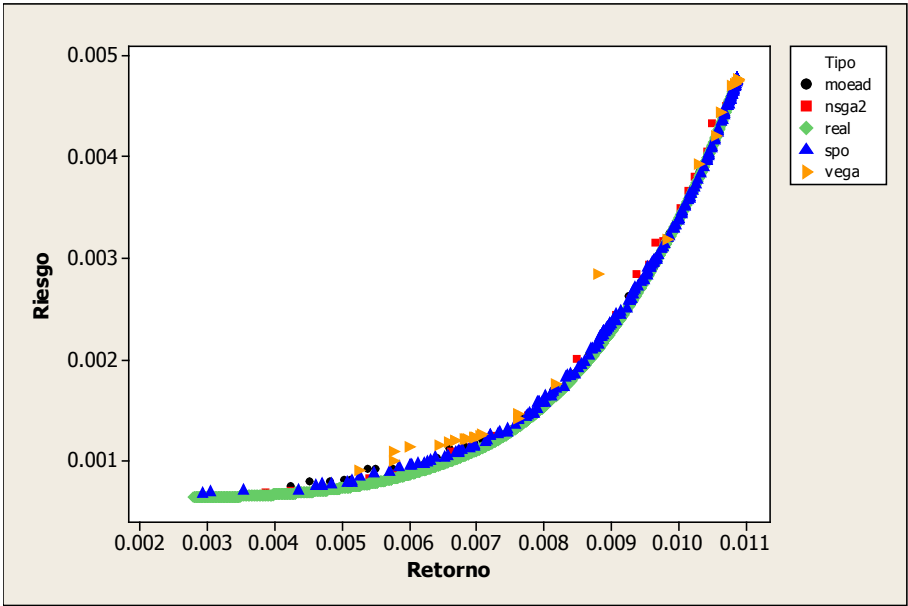


Tabla 4.8 HV_{perc} inst0 mediana

estrategia	tiempo	hv_percent
moead	9.386440039	97.6147%
nsga2	10.27579498	96.5479%
spo	8.684428215	98.2979%
vega	9.466565847	89.2091%

Figura 4.11 Inst0 mediana (b)

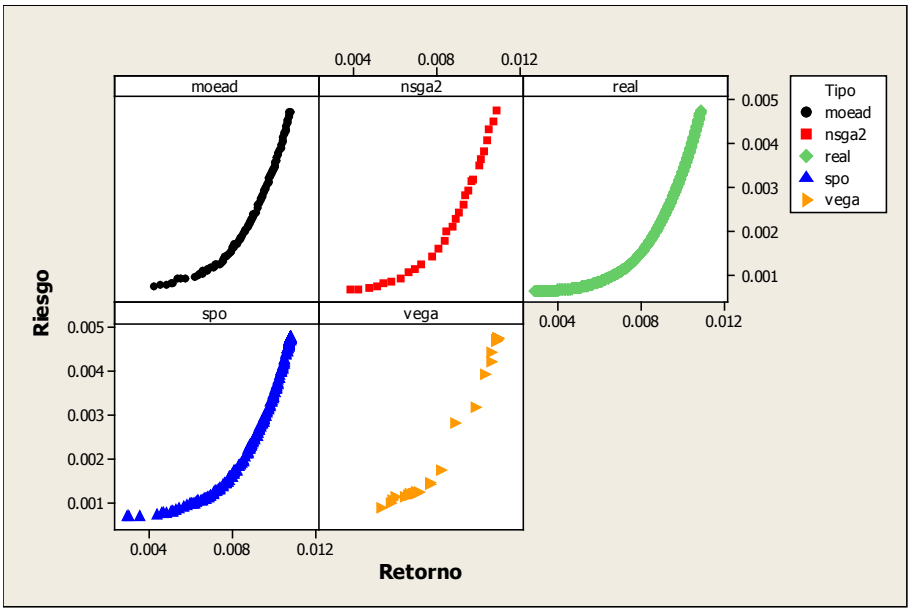


Figura 4.12 Inst1 mediana (a)

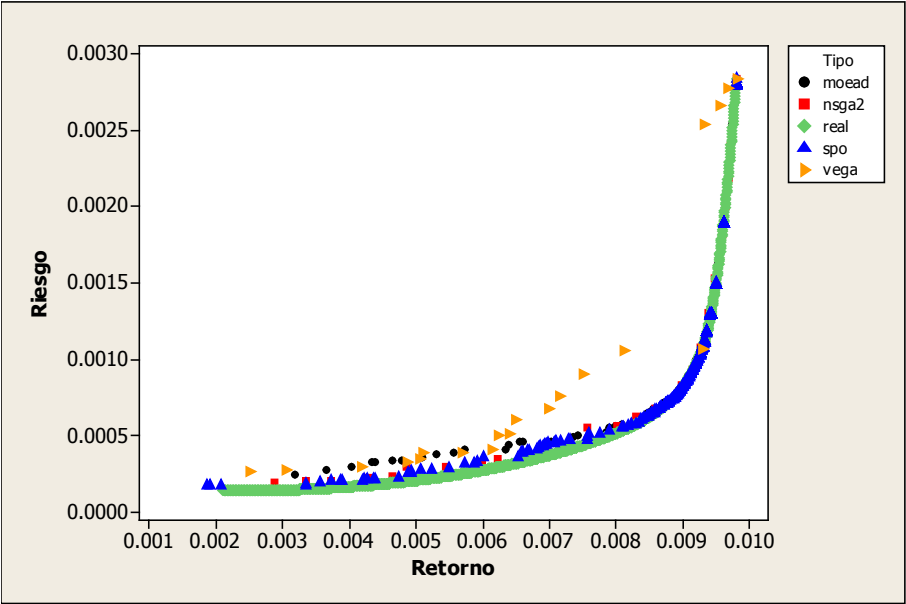


Tabla 4.9 Métricas inst1 mediana

estrategia	tiempo	hv_percent
moead	137.469667	95.8584%
nsga2	139.918093	97.0631%
spo	134.3552189	97.6835%
vega	137.5779932	87.5550%

Figura 4.13 Inst1 mediana (b)

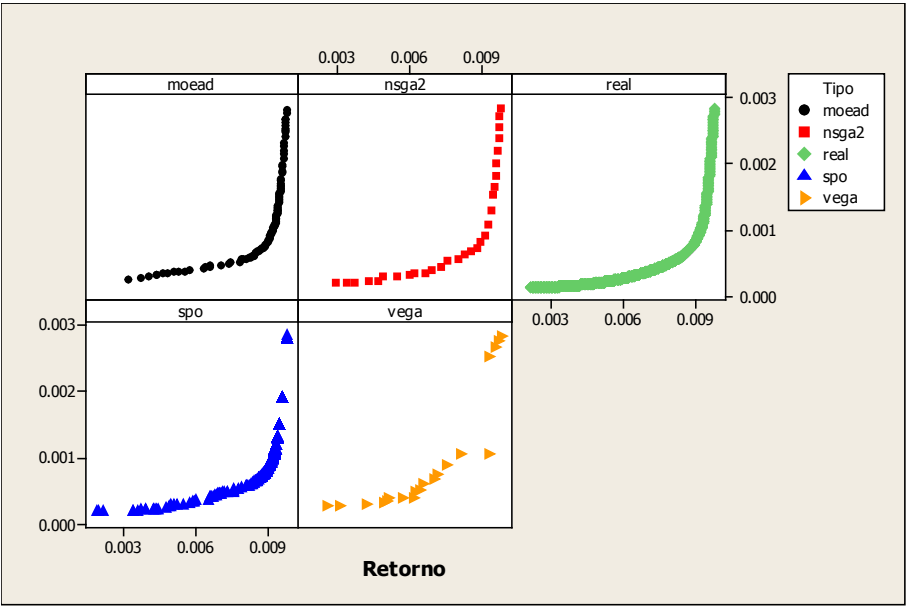


Figura 4.14 Inst2 mediana (a)

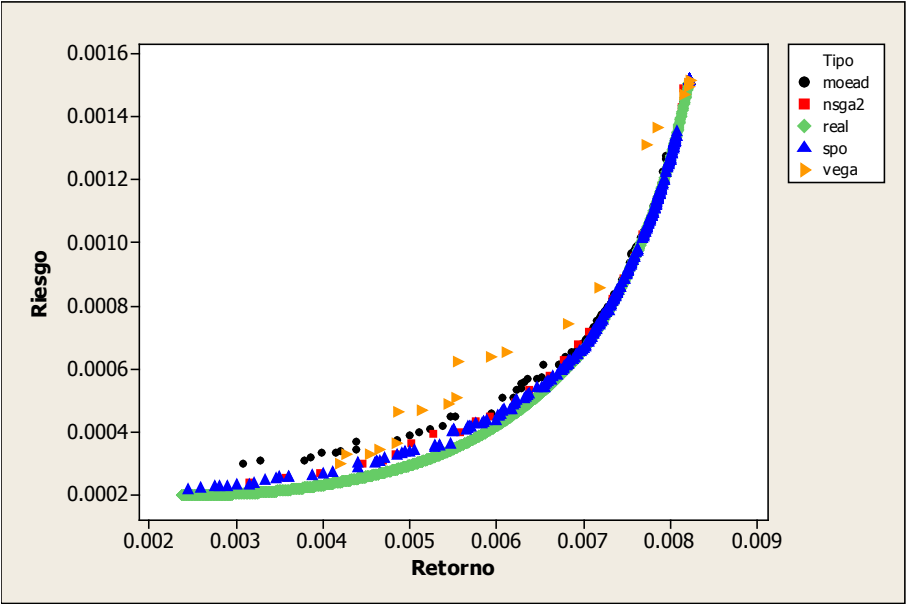


Tabla 4.10 Métricas inst2 mediana

estrategia	tiempo	hv_percent
moead	153.8806272	92.7641%
nsga2	150.2436111	94.9682%
spo	154.263298	97.2482%
vega	154.0000622	82.8718%

Figura 4.15 Inst2 mediana (b)

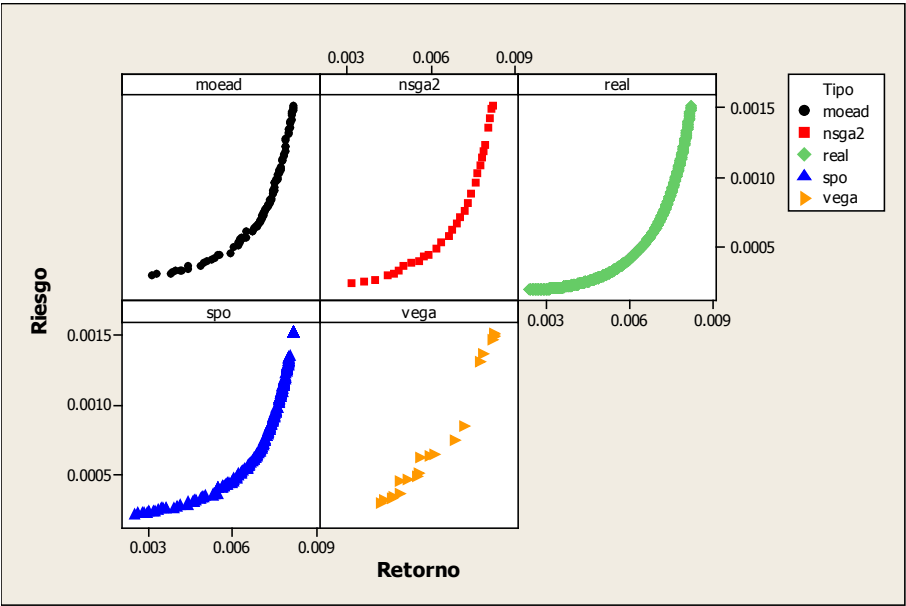


Figura 4.16 Inst3 mediana (a)

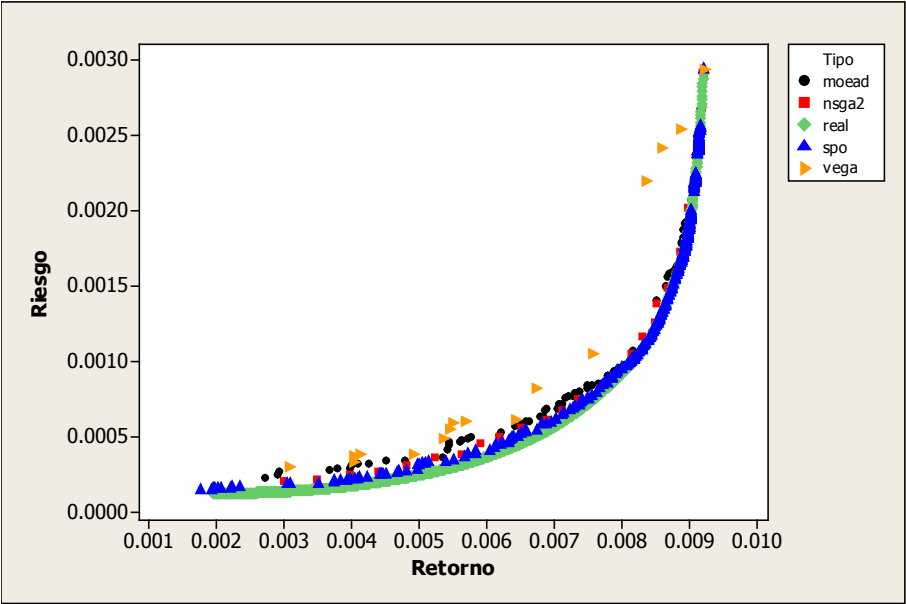


Tabla 4.11 Métricas inst3 mediana

estrategia	tiempo	hv_percent
moead	201.986851	95.0938%
nsga2	198.9523768	95.7831%
spo	204.486424	98.1723%
vega	202.1144722	81.3964%

Figura 4.17 Inst3 mediana (b)

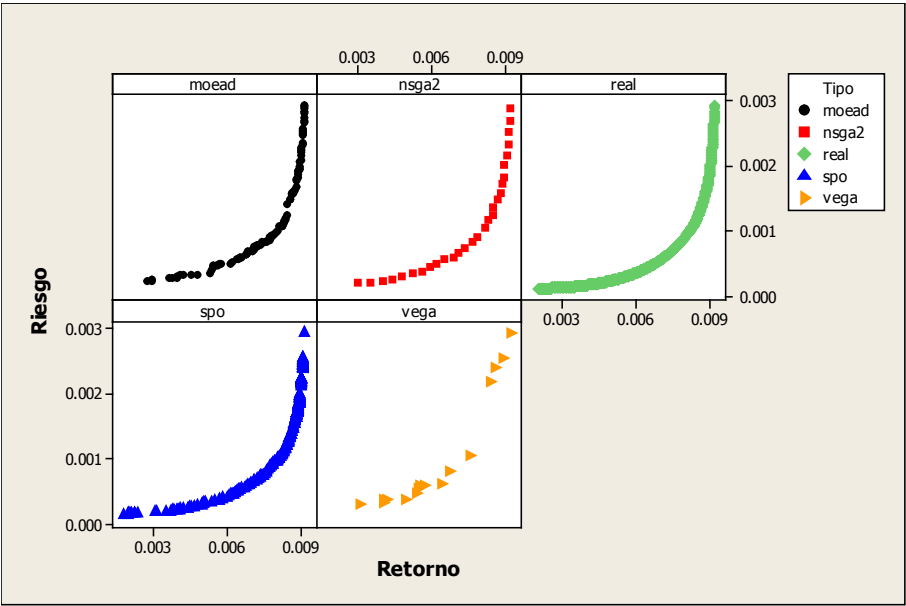


Figura 4.18 Inst4 mediana (a)

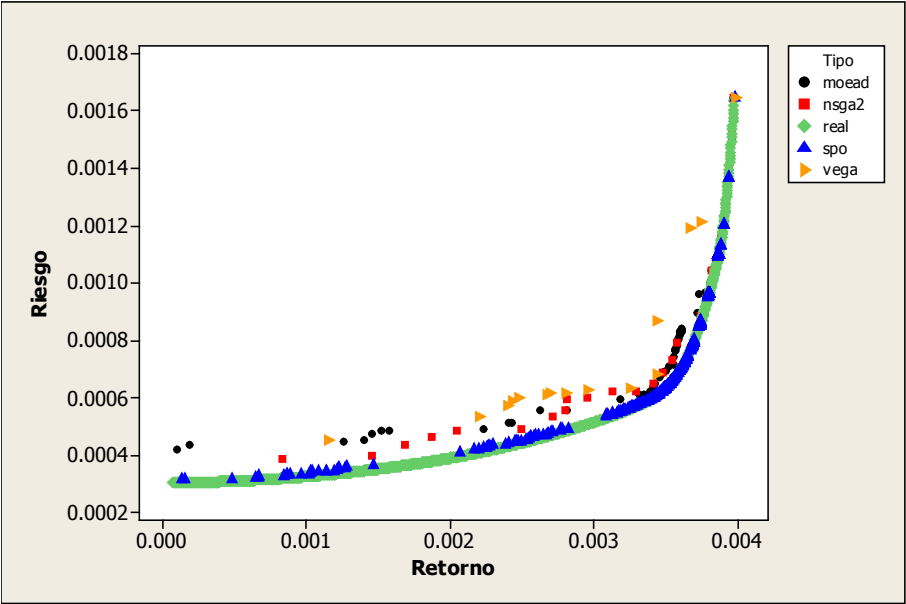
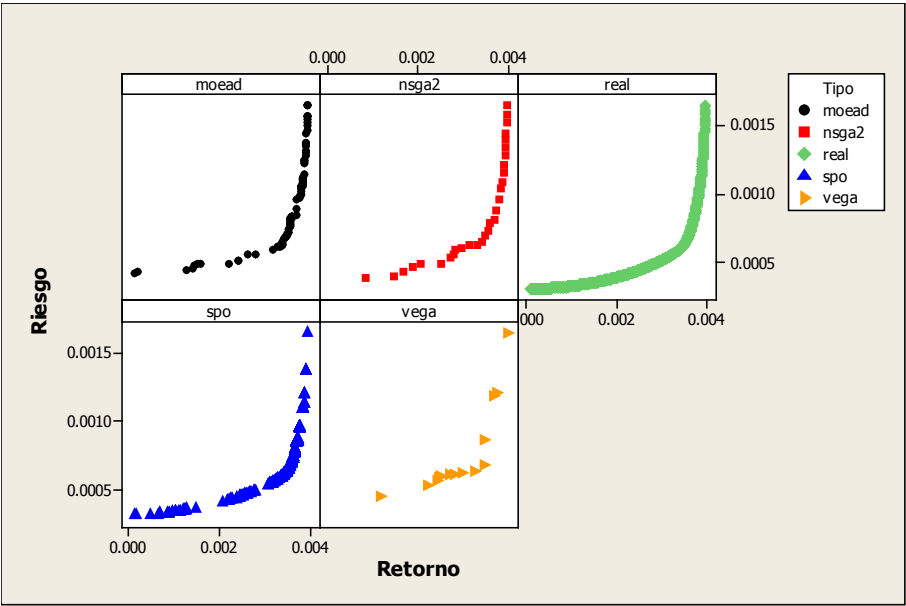


Tabla 4.12 Métricas inst4 mediana

estrategia	tiempo	hv_percent
moead	2398.413085	91.4709%
nsga2	2289.414533	93.4996%
spo	2605.19272	98.4124%
vega	2406.257872	83.4054%

Figura 4.19 Inst4 mediana (b)



En la Tabla 4.13 se muestra el resumen estadístico con respecto a cada métrica. Se refiere al lector al apéndice Estadísticos descriptivos de implementación final tiempo ajustado para conocer los estadísticos descriptivos completos.

Tabla 4.13 Resumen estadístico resultados finales

Instancia	estrategia	HV_perc			gd			ESS			card		
		Mejor	Mediana	Peor	Mejor	Mediana	Peor	Mejor	Mediana	Peor	Mejor	Mediana	Peor
inst0	moead	98.05%	97.60%	96.63%	1.69E-05	3.43E-05	5.46E-05	4.06E-05	5.96E-05	2.14E-04	100	100	100
	nsga2	96.98%	96.54%	95.49%	3.42E-05	4.62E-05	7.19E-05	6.81E-05	1.18E-04	1.51E-04	30	30	28
	spo	98.56%	98.29%	98.07%	1.39E-05	1.99E-05	2.93E-05	3.56E-05	4.67E-05	8.85E-05	317	266.5	211
	vega	92.38%	88.81%	82.10%	1.09E-04	1.49E-04	1.97E-04	1.64E-04	3.10E-04	5.06E-04	41	32	21
Inst1	moead	96.20%	95.84%	95.38%	2.41E-05	3.56E-05	5.29E-05	4.59E-05	8.64E-05	1.91E-04	100	100	99
	nsga2	97.56%	97.04%	96.42%	2.54E-05	3.05E-05	4.29E-05	6.72E-05	1.06E-04	1.73E-04	30	30	29
	spo	97.94%	97.67%	96.78%	3.44E-06	5.03E-06	6.35E-06	1.80E-05	3.39E-05	4.78E-05	1080	1008	887
	vega	90.84%	87.51%	79.83%	1.23E-04	2.03E-04	3.57E-04	1.86E-04	2.73E-04	5.96E-04	28	21.5	15
Inst2	moead	93.70%	92.76%	91.74%	2.07E-05	3.29E-05	4.29E-05	2.80E-05	7.06E-05	1.80E-04	100	100	96
	nsga2	95.39%	94.96%	94.42%	2.31E-05	2.84E-05	3.32E-05	5.15E-05	7.98E-05	1.37E-04	30	30	29
	spo	97.52%	97.22%	96.99%	5.97E-06	7.61E-06	1.16E-05	1.53E-05	1.98E-05	3.50E-05	542	496.5	364
	vega	85.53%	82.86%	74.32%	8.03E-05	1.17E-04	1.58E-04	1.20E-04	1.81E-04	3.72E-04	23	19	11
Inst3	moead	95.51%	95.09%	94.70%	5.80E-05	7.12E-05	9.17E-05	3.57E-05	7.14E-05	1.72E-04	100	100	91
	nsga2	96.48%	95.78%	94.90%	4.39E-05	5.85E-05	7.76E-05	7.12E-05	9.71E-05	1.47E-04	30	30	29
	spo	98.45%	98.17%	97.98%	9.15E-06	1.07E-05	1.30E-05	1.68E-05	1.97E-05	2.72E-05	709	625	505
	vega	83.84%	81.35%	78.92%	2.29E-04	3.15E-04	4.48E-04	1.13E-04	2.09E-04	4.02E-04	35	30	16
Inst4	moead	93.11%	91.39%	89.82%	3.02E-05	4.49E-05	7.56E-05	4.02E-05	8.63E-05	1.58E-04	100	80.5	62
	nsga2	94.23%	93.47%	92.30%	2.34E-05	4.01E-05	6.63E-05	3.05E-05	8.03E-05	2.93E-04	30	30	28
	spo	98.75%	98.41%	97.99%	1.89E-06	2.28E-06	3.63E-06	1.31E-05	1.55E-05	2.90E-05	867	750.5	665
	vega	85.90%	83.34%	77.90%	9.33E-05	1.17E-04	1.69E-04	7.72E-05	1.58E-04	3.98E-04	23	16	11

4.4 Análisis empírico global

En la Tabla 4.14 se muestra una clasificación de los resultados con respecto a cada métrica de la siguiente manera: en cada columna, con el nombre de la métrica se muestra en parejas la estrategia con la respectiva instancia que

resolvió, tales parejas están ordenadas en forma descendente por su desempeño con respecto a la mediana de cada métrica³². Por ejemplo, respecto a gd , el mejor valor para la mediana es para SPO al resolver inst4 (pareja en negritas), en segundo puesto (siguiente fila) se encuentra también SPO pero al resolver inst1, y así sucesivamente hasta que, con respecto a gd , la peor mediana es la última pareja de la columna (color rojo), y se trata de VEGA resolviendo inst3.

Tabla 4.14 Rankin con respecto a la mediana de cada métrica (mayor a menor desempeño)

Posición	HV_{perc}		gd		ESS		$card$	
	estrategia	instancia	estrategia	instancia	estrategia	instancia	estrategia	instancia
1	SPO	inst4	SPO	inst4	SPO	inst4	SPO	inst1
2	SPO	inst0	SPO	inst1	SPO	inst3	SPO	inst4
3	SPO	inst3	SPO	inst2	SPO	inst2	SPO	inst3
4	SPO	inst1	SPO	inst3	SPO	inst1	SPO	inst2
5	MOEA/D	inst0	SPO	inst0	SPO	inst0	SPO	inst0
6	SPO	inst2	NSGA-II	inst2	MOEA/D	inst0	MOEA/D	inst0
7	NSGA-II	inst1	NSGA-II	inst1	MOEA/D	inst2	MOEA/D	inst2
8	NSGA-II	inst0	MOEA/D	inst2	MOEA/D	inst3	MOEA/D	inst3
9	MOEA/D	inst1	MOEA/D	inst0	NSGA-II	inst2	MOEA/D	inst1
10	NSGA-II	inst3	MOEA/D	inst1	NSGA-II	inst4	MOEA/D	inst4
11	MOEA/D	inst3	NSGA-II	inst4	MOEA/D	inst4	VEGA	inst0
12	NSGA-II	inst2	MOEA/D	inst4	MOEA/D	inst1	NSGA-II	inst2
13	NSGA-II	inst4	NSGA-II	inst0	NSGA-II	inst3	NSGA-II	inst4
14	MOEA/D	inst2	NSGA-II	inst3	NSGA-II	inst1	NSGA-II	inst3
15	MOEA/D	inst4	MOEA/D	inst3	NSGA-II	inst0	NSGA-II	inst1
16	VEGA	inst0	VEGA	inst2	VEGA	inst4	NSGA-II	inst0
17	VEGA	inst1	VEGA	inst4	VEGA	inst2	VEGA	inst3
18	VEGA	inst4	VEGA	inst0	VEGA	inst3	VEGA	inst1
19	VEGA	inst2	VEGA	inst1	VEGA	inst1	VEGA	inst2
20	VEGA	inst3	VEGA	inst3	VEGA	inst0	VEGA	inst4

³² HV_{perc} y $card$ de mayor a menor magnitud, gd y ESS de menor a mayor

La anterior tabla se presenta con la intención de concluir que, después de analizar bajo el criterio hipervolumen, en general con respecto a las demás métricas (*gd*, *ESS* y *card*) el desempeño comparado de las diferentes técnicas muestra una tendencia muy similar. Sin importar la instancia tratada, SPO obtiene el mejor desempeño para todas las métricas. Después vienen el NSGA-II y el MOEA/D, compartiendo los rangos intermedios con resultados similares para las diferentes métricas. Finalmente, se confirma el mal desempeño de VEGA, que ocupa los últimos lugares para todas las métricas.

4.5 Comparación con el estado del arte

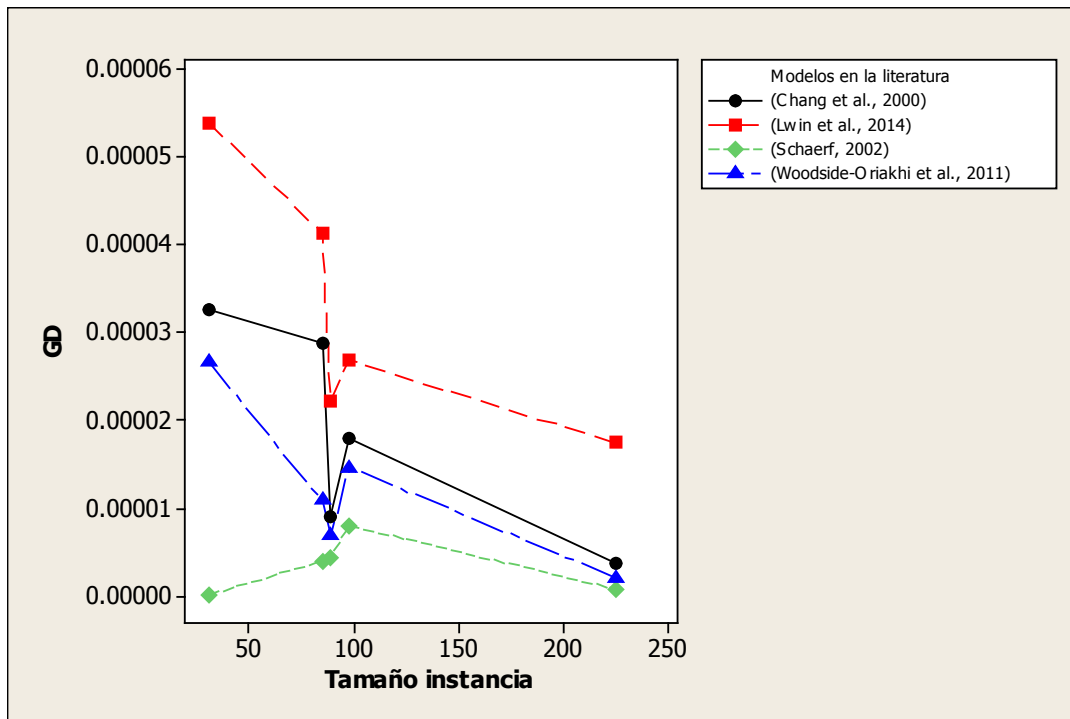
En la Tabla 4.15 se muestran los resultados obtenidos por los modelos más representativos (Jin, Qu, & Atkin, 2016). La métrica presentada es distancia generacional (*gd*), debido a que los autores únicamente reportan ésta. Es preciso mencionar que, pese a que en la presente investigación es de gran relevancia el hipervolumen, la distancia generacional tiene tendencias muy similares de acuerdo a los análisis que se han presentado en el transcurso del presente documento.

Tabla 4.15 Estado del arte, fuente: (Jin, Qu, & Atkin, 2016)

Modelos en la literatura	Instancia	Tamaño	gd
(Chang et al., 2000)	Inst0	31	3.25E-05
	Inst1	85	2.87E-05
	Inst2	89	9.08E-06
	Inst3	98	1.80E-05
	Inst4	225	3.72E-06
(Schaerf, 2002)	Inst0	31	8.86E-08
	Inst1	85	3.92E-06
	Inst2	89	4.39E-06
	Inst3	98	7.97E-06
	Inst4	225	7.25E-07
(Lwin et al., 2014)	Inst0	31	5.38E-05
	Inst1	85	4.14E-05
	Inst2	89	2.22E-05
	Inst3	98	2.68E-05
	Inst4	225	1.74E-05
(Woodside-Oriakhi et al., 2011)	Inst0	31	2.66E-05
	Inst1	85	1.08E-05
	Inst2	89	6.81E-06
	Inst3	98	1.46E-05
	Inst4	225	2.06E-06

La Figura 4.20 muestra los datos de la Tabla 4.15 de manera gráfica, que permiten establecer un ranquin general para todas las instancias, el cual, de acuerdo al desempeño sobre la métrica presentada, de mejor a peor calidad para todas las instancias es: Schaerf (2002), Woodside-Oriakhi et al. (2011), Chang et al. (2000) y Lwin et al. (2014). De acuerdo a este orden se ha hecho el comparativo con respecto a los resultados de la presente investigación.

Figura 4.20 Estado del arte, Distancia Generacional (gd)



En la Tabla 4.16 se muestran los resultados de la literatura comparados con los de la técnica que obtuvo los mejores resultados en la presente investigación (los cuales fueron obtenidos por SPO, revisar sección 4.3 y 4.4). La columna *diferencia* es el valor obtenido menos el valor del estado del arte³³, mientras que la columna *diferencia porcentual* es el mismo valor representado como proporción del valor obtenido, e.g. una *diferencia porcentual* de 99.56% significa que el valor del estado del arte es un 99.56% menor que el obtenido en el presente documento. La fórmula para el cálculo de la diferencia porcentual es la siguiente:

³³ Si es negativo la técnica propuesta supera a la del estado del arte en calidad de solución, si es positivo será lo contrario

$$\Delta GD\% = \frac{\text{valor obtenido} - \text{estado del arte}}{\text{valor obtenido}} \cdot 100\%.$$

Tabla 4.16 Comparativo con el estado del arte

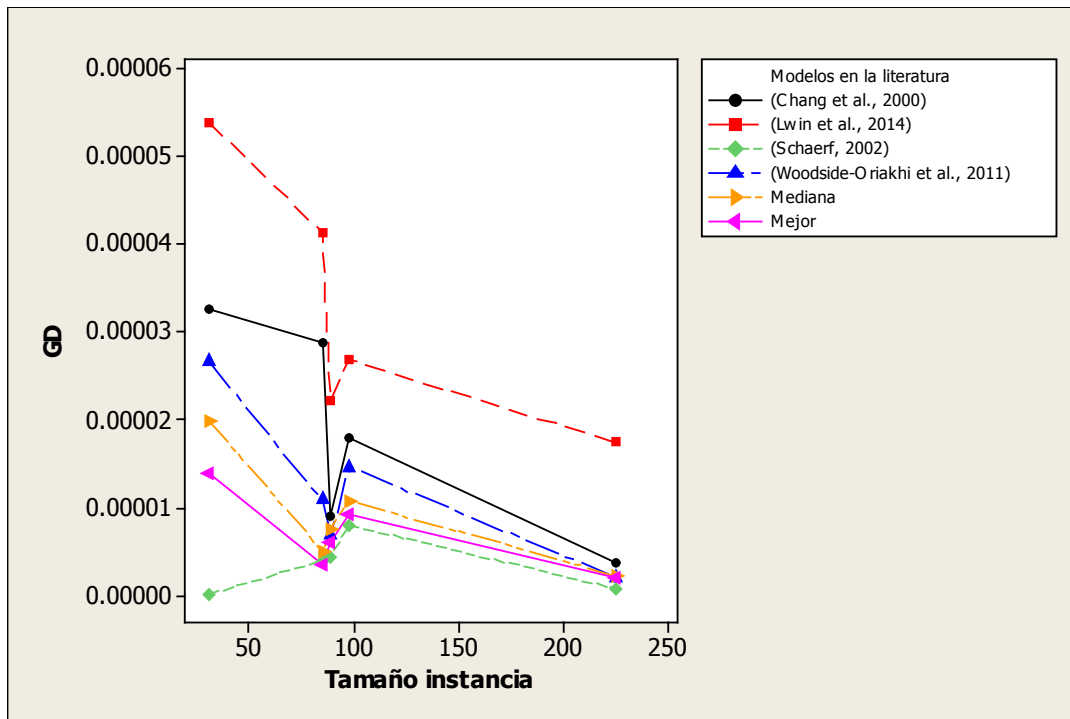
Modelo en la literatura				Modelo en la investigación						
	Instancia	Tamaño	GD	Estrategia	Mediana (GD)	Diferencia	Diferencia Porcentual	Mejores (GD)	Diferencia	Diferencia Porcentual
(Schaerf, 2002)	Inst0	31	8.86E-08	spo	1.99E-05	1.98E-05	99.56%	1.39E-05	1.38E-05	99.36%
	Inst1	85	3.92E-06	spo	5.00E-06	1.08E-06	21.59%	3.44E-06	-4.81E-07	-14.00%
	Inst2	89	4.39E-06	spo	7.55E-06	3.16E-06	41.87%	5.97E-06	1.58E-06	26.49%
	Inst3	98	7.97E-06	spo	1.07E-05	2.74E-06	25.60%	9.15E-06	1.18E-06	12.89%
	Inst4	225	7.25E-07	spo	2.24E-06	1.52E-06	67.65%	1.89E-06	1.17E-06	61.71%
(Woodside-Oriakhi et al., 2011)	Inst0	31	2.66E-05	spo	1.99E-05	-6.67E-06	-33.48%	1.39E-05	-1.27E-05	-91.66%
	Inst1	85	1.08E-05	spo	5.00E-06	-5.80E-06	-116.02%	3.44E-06	-7.36E-06	-214.07%
	Inst2	89	6.81E-06	spo	7.55E-06	7.42E-07	9.82%	5.97E-06	-8.38E-07	-14.04%
	Inst3	98	1.46E-05	spo	1.07E-05	-3.89E-06	-36.29%	9.15E-06	-5.45E-06	-59.58%
	Inst4	225	2.06E-06	spo	2.24E-06	1.81E-07	8.09%	1.89E-06	-1.66E-07	-8.79%
(Chang et al., 2000)	Inst0	31	3.25E-05	spo	1.99E-05	-1.26E-05	-63.08%	1.39E-05	-1.86E-05	-134.17%
	Inst1	85	2.87E-05	spo	5.00E-06	-2.37E-05	-474.05%	3.44E-06	-2.53E-05	-734.62%
	Inst2	89	9.08E-06	spo	7.55E-06	-1.53E-06	-20.24%	5.97E-06	-3.11E-06	-52.05%
	Inst3	98	1.80E-05	spo	1.07E-05	-7.29E-06	-68.03%	9.15E-06	-8.85E-06	-96.74%
	Inst4	225	3.72E-06	spo	2.24E-06	-1.48E-06	-65.97%	1.89E-06	-1.83E-06	-96.45%
(Lwin et al., 2014)	Inst0	31	5.38E-05	spo	1.99E-05	-3.39E-05	-169.96%	1.39E-05	-3.99E-05	-287.64%
	Inst1	85	4.14E-05	spo	5.00E-06	-3.64E-05	-728.07%	3.44E-06	-3.80E-05	-1103.94%
	Inst2	89	2.22E-05	spo	7.55E-06	-1.46E-05	-193.98%	5.97E-06	-1.62E-05	-271.76%
	Inst3	98	2.68E-05	spo	1.07E-05	-1.61E-05	-150.18%	9.15E-06	-1.77E-05	-192.92%
	Inst4	225	1.74E-05	spo	2.24E-06	-1.52E-05	-676.32%	1.89E-06	-1.55E-05	-818.87%

La comparación se ha realizado con las medianas obtenidas y también con los mejores valores.

Con respecto a las medianas, de acuerdo a la columna de *Diferencia Porcentual* correspondiente de la Tabla 4.16, en ninguna instancia se alcanzan o superan los resultados de Schaerf. A los resultados de Woodside-Oriakhi et al. se les supera en 3 de 5 instancias. Finalmente, con respecto a las otras dos técnicas usadas para la comparación, en ninguna instancia se obtiene una Diferencia Porcentual mayor al -20% del valor obtenido por SPO, o sea que el algoritmo genético con sumas ponderadas (SPO) produce siempre mejores resultados con respecto a las técnicas de Chang et al. (2000) y Lwin et al. (2014).

Con respecto a los mejores resultados obtenidos, un análisis similar demuestra que se logra superar a Schaerf solo en inst1. Por otro lado, se supera a todos los modelos restantes. La información de la Tabla 4.16 se puede ver resumida en la Figura 4.21, donde las series nombradas *mejor* y *mediana* son los obtenidos por el algoritmo SPO.

Figura 4.21 Comparación con el estado del arte (*gd*)



Cabe mencionar que la métrica *gd* está expresada en una magnitud muy pequeña y además existe una relación inversa entre el valor de la métrica y la dificultad para mejorarla³⁴. Por lo anterior, la variación porcentual es una buena medida de la diferencia de eficiencia entre los algoritmos, dado que, a medida que el resultado (v_0) es más cercano a cero, la diferencia con respecto a la

³⁴ A medida que *GD* es menor es más difícil mejorar la solución (obtener un valor menor)

cantidad con la que será comparada (v_f), representará un porcentaje mucho mayor del valor obtenido, e.g. supóngase que se tiene $v_0 = 0.5$ y $v_f = 0.501$ éstos son resultados alejados entre sí en 0.001 y $\Delta v\% = \frac{0.5-0.501}{0.5} = -0.2\%$; por otro lado, en un segundo caso se tiene $v_0 = 0.049$ y $v_f = 0.05$ los cuales son resultados igualmente alejados por 0.001, pero $\Delta v\% = \frac{0.049-0.05}{0.049} = -2.04\%$. A pesar de estar separados los resultados de los dos casos en igual magnitud, es más difícil pasar de 0.05 a 0.049 que hacerlo de 0.501 a 0.5 por lo tanto se valora más el trabajo del algoritmo en el segundo caso y la diferencia porcentual lo considera al corresponderle un $|\Delta v\%|$ mayor.

Conclusiones

En el presente proyecto se estudió, adaptó, implementó un conjunto de metaheurísticas conocidas en el estado del arte y se analizaron sus resultados de aplicarlas a un problema financiero clásico. Se hizo frente a múltiples etapas de análisis a lo largo de la obtención de datos experimentales. En el transcurso de tales etapas, la información fue abriendo pauta a indagaciones acerca de las características más relevantes como el tipo de paradigma en el que se basan las metaheurísticas, o la convexidad y forma de los frentes de Pareto reales. Se han señalado algunos efectos en las variables, que repercuten en el funcionamiento eficiente de los algoritmos, así como plantear explicaciones acerca de las posibles causas que llevaron a ello, lo cual funge como el valor más relevante de la presente investigación.

En el análisis de los métodos de selección de poblaciones no dominadas se logró destacar, acerca de la relación entre comparaciones de dominancia y tiempo de ejecución, que no necesariamente guardan una relación directamente proporcional, es decir, un mayor uso de la operación de comparación, no implica necesariamente un mayor tiempo de ejecución (esto probablemente se debe a que, en el problema estudiado el número de objetivos es mínimo). En cuanto a la relación entre llamadas a la función objetivo para las metaheurísticas y el tiempo de ejecución, en la investigación fue ineludible tomar en cuenta la relación entre ambos factores. En el entorno multiobjetivo las operaciones de clasificación de Pareto constituyen un elemento de gran

relevancia, y son particularmente costosas aun tratándose de dos objetivos. Fue necesario mantener un límite de población bajo para el NSGA-II, para que el tiempo de ejecución no se elevara ($O(\Psi \log^{m-1} \Psi)$). En SPO y MOEA/D en un menor grado aunque no se eligieron tamaños de población grandes. Por otra parte el tamaño de población fue un parámetro que no se relacionó con el tiempo en VEGA.

El método exhaustivo para el ajuste de parámetros, a pesar de tener un costo computacional elevado, fue posible implementarlo ejecutando un número relativamente bajo de pruebas para poder llegar a una configuración que garantizara buenos resultados. Analizar valores extremos para los parámetros (sección 3.3.1) fue interesante debido a que, no solo se delimitó un intervalo de búsqueda, sino que se examinaron valores poco convencionales en la práctica que, en algunos casos, pasaron a ser adecuados.

Sería una labor interminable explicar en detalle el comportamiento de las métricas o tan solo garantizar si tienen alguna relación entre ellas. Pero en el área de las metaheurísticas, descartar situaciones que deterioran el funcionamiento de una forma muy marcada, permite aumentar las posibilidades de obtener soluciones buenas.

Se observó, en el transcurso de la investigación, que algunas veces es adecuado determinar si la heurística está utilizando tiempo innecesario (a partir de cierto tiempo ya no se puede mejorar, y no por la presencia de un óptimo local). Es decir, hay veces que el MOEA/D alcanza $HV_{perc} = 95\%$ en 10

segundos y, a partir de ese momento, ya no hay más mejoras significativas. O sea que en 10 segundos MOEA/D ya dio el máximo rendimiento y ya no tiene sentido ejecutarlo por más tiempo. En un estudio posterior, podría estudiarse más a fondo este límite de eficiencia, el cual es muy distinto en cada estrategia.

No se esperaba que un Algoritmo Genético basado en una Suma Ponderada de Objetivos se mostrara como el de mayor eficiencia. Esto se debe, posiblemente a dos factores, el primero, el número de objetivos es bajo, y, por otro lado, los Frentes reales son muy uniformes y sin discontinuidades³⁵.

Analizando la forma operativa de cada algoritmo, se abre pauta a concluir que hay factores dentro de éstos, que los hacen mejores al manejar las características particulares de cada instancia resuelta, la presente investigación abrió pauta al estudio a detalle de este aspecto como una perspectiva a futuro, inclusive cada estrategia que fue implementada, es sensible de diferente forma a cada uno de los parámetros calibrados. Aunque el tamaño de la instancia es crucial en la dificultad del problema, otras características del frente de Pareto (ver Figura 1.2 y sección de implementación) también tendrán relevancia en la eficacia con la que particularmente cada estrategia resuelva el problema, y ese es un factor que no hay que perder de vista al elegir una técnica para resolver un MOP.

³⁵ E.g. de acuerdo a la experimentación en el estado del arte (Coello, Van-Veldhuizen, & Lamont, 2002), se ha identificado que el NSGA-II presenta una desventaja al trabajar con frentes discontinuos, mientras que con MOEA/D no es así.

Cabe mencionar, que tampoco la diferencia de desempeño entre estrategias fue tan significativa (Capítulo 1). En general los resultados obtenidos son satisfactorios para MOEA/D, NSGA-II y SPO. La estrategia VEGA fue de las primeras propuestas para MOPs, debido a esto, algoritmos más recientes logran superarla fácilmente. Por otro lado el haber estudiado esta técnica, abre pautas a hacer una retroalimentación de su funcionamiento.

Bibliografía

- Beasley, J. E. (1990). *OR-Library*. Recuperado el 2015, de <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html>
- Bentley, J. L., Kennet, L. C., & David, B. L. (1993). Fast linear expected-time algorithms for computing maxima and convex hulls. *Algorithmica*, 168-183.
- Chang, T. J., Meade, N., Beasley, J. E., & Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*(27), 1271 - 1302.
- Coello, C. A. (Enero de 2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12), 1245-1287.
- Coello, C. A. (2008). *Introducción a la Computación Evolutiva*. México, D.F.: CINVESTAV-IPN.
- Coello, C. A., Van-Veldhuizen, D. A., & Lamont, G. B. (Mayo de 2002). *Evolutionary Algorithms for Solving Multi-Objective Problems* (1era ed.). New York: Kluwer Academic Publishers.
- Collette, Y., & Siarry, P. (2004). *Multiobjective Optimization: Principles and Case Studies*. New York: Springer-Verlag Berlin Heidelberg GmbH.
- Darwin, C. (1859). *On the origin of species by means of natural selection, or, the preservation of favoured races in the struggle for life*. London: J. Murray.
- Deb, K., & Bhushan Agrawal, R. (1995). Simulated Binary Crossover for Continuous Search Space. *Complex Systems*(9), 115 - 148.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (02 de Abril de 2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, 182 - 197.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning* (1st ed.). Boston, USA: Addison-Wesley Longman Publishing Co.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Michigan, USA: Ann Arbor.
- Jin, Y., Qu, R., & Atkin, J. (Febrero de 2016). Constrained Portfolio Optimisation: the state-of-the-art Markowitz Models. *The 2016 International Conference on Operations Research and Enterprise Systems* (págs. 388-395). Roma: University of Nottingham, UK, Faculty of Science, School of Computer Science.

- Lomelí, H., & Rumbos, B. (2001). *Métodos Dinámicos en Economía. Otra Búsqueda del Tiempo Perdido*. México D.F.: Instituto Tecnológico Autónomo de México.
- Lwin, K., Qu, R., & Kendall, G. (2014). A learning guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing*(24), 757 – 772.
- Markowitz, H. (Marzo de 1952). Portfolio Selection. *The Journal of Finance*, 7(1), 77 - 91.
- Mezura Montes, E. (2001). *Uso de la Técnica Multiobjetivo NPGA para el Manejo de Restricciones en Algoritmos Genéticos (Tesis de grado de Maestro)*. Xalapa, Veracruz, México.
- Michalewicz, Z., & Attia, N. F. (1994). Evolutionary optimization of constrained problems. (A. V. Sebald, Ed.) *Third annual conference of evolutionary Programming*, 98-108.
- Ponsich, A., López Jaimes, A., & Coello, C. A. (June de 2013). A Survey on Multiobjective Evolutionary Algorithms for the Solution of the Portfolio Optimization Problem and Other Finance and Economics Applications. *IEEE Transactions on Evolutionary Computation*, 17(3), 321 - 344.
- Schaerf, A. (2002). Local search techniques for constrained portfolio selection problems. *Computational Economics*(20), 177–190.
- Schaffer, J. D. (1985). Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms* (págs. 93–100). New Jersey, USA: Lawrence Erlbaum.
- Schott, J. R. (1995). *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*. Massachusetts: Massachusetts Institute of Technology, Cambridge.
- Srinivas, N., & Deb, K. (1995). Multiobjective function optimization using nondominated sorting genetic algorithms. *Evol. Comput.*, 2(3), 221–248.
- Van-Veldhuizen, D. A. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Ohio: Air Force Institute of Technology.
- Woodside-Oriakhi, M., Lucas, C., & Beasley, J. (2011). Heuristic algorithms for the cardinality constrained efficient frontier. *European Journal of Operational Research*(213(3)), 538 – 550.
- Zhang, Q., & Li, H. (Diciembre de 2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6), 712 - 731.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., & Grunert da Fonseca, V. (Abril de 2003). Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, 7(2), 117 - 132.

Apéndices

Interpretación de resultados en las pruebas de hipótesis

En el presente proyecto se han realizado múltiples pruebas para decidir si existen diferencias estadísticas entre resultados obtenidos. En este apéndice se proporciona una breve explicación de cómo interpretar y utilizar la información obtenida.

Prueba de rangos de signos de Wilcoxon

Se trata de una prueba no paramétrica en la que se utiliza información con respecto al signo y magnitud de las diferencias en pares de datos de dos muestras. El único supuesto que exige tal prueba es que las muestras provengan de distribuciones continuas.

Sea $(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)$ observaciones muestrales pareadas, se intenta probar si las variables difieren entre sí. La hipótesis probada es entonces $H_0: p = 0.5$, donde p es la probabilidad que la variable x es mayor a y , es decir, se prueba si las distribuciones son idénticas ya que la probabilidad de que y sea mayor que x sería la misma ($1 - p = 0.5$). El estadístico calculado para la prueba tiene una distribución aproximadamente normal $N(0,1)$.

Una de las decisiones de interés en la presente investigación es determinar si una configuración de parámetros es mejor a otra con respecto a una métrica.

Por ejemplo, en la Tabla 0.1 se muestra un resultado de una prueba, donde se compara un conjunto de pruebas de un algoritmo utilizando un tamaño de población (NP) de 500 y otro de 1000. Se intenta determinar si es preferible utilizar alguna de las dos configuraciones de acuerdo a la métrica HV_{perc} con un nivel de significancia de $\alpha = 0.05$.

Tabla 0.1 Ejemplo: Prueba de rangos de signos de Wilcoxon

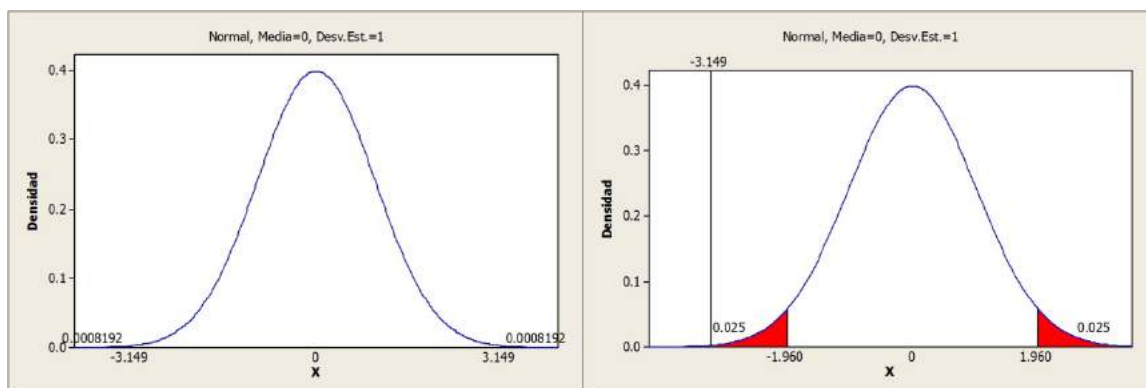
Two-sample wilcoxon rank-sum (Mann-whitney) test			
np	obs	rank sum	expected
500	60	3030	3630
1000	60	4230	3630
combined	120	7260	7260
unadjusted variance	36300.00		
adjustment for ties	0.00		
adjusted variance	36300.00		
Ho: hv_per~t(np==500) = hv_per~t(np==1000)			
z = -3.149			
Prob > z = 0.0016			

De acuerdo a la Tabla 0.1, los resultados corresponden a una prueba de dos colas:

$$H_0: HV_{perc}(NP = 500) = HV_{perc}(NP = 1000)$$

$$H_1: HV_{perc}(NP = 500) \neq HV_{perc}(NP = 1000)$$

Figura 0.1 Ejemplo: gráfica de distribuciones normales (a)



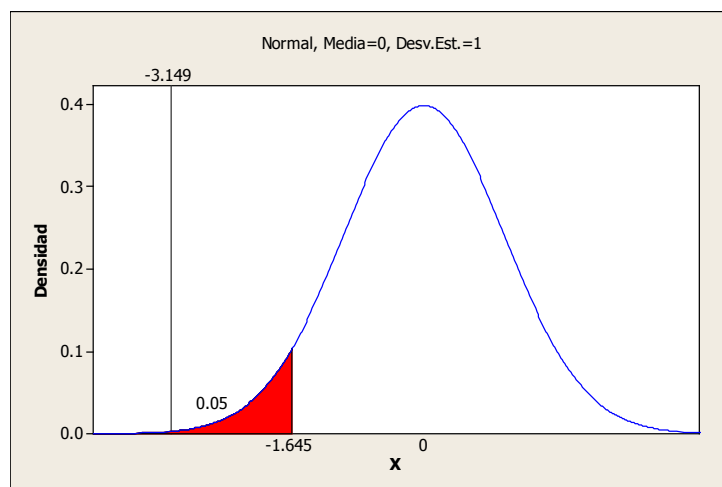
Es decir, dado que $0.0016 < \alpha$ el estadístico z se ubica en zona de rechazo (Figura 0.1) y podemos solo concluir que las variables son diferentes, pero no cuál es mayor o menor de ellas. Por otro lado, el estadístico $z = -3.149$ puede dar más información con respecto a ello (note que $p(z < -3.149) = \frac{0.0016}{2} = 0.0008$). Recuerde que a una prueba de cola izquierda le corresponden:

$$H_0: HV_{perc}(NP = 500) \geq HV_{perc}(NP = 1000)$$

$$H_1: HV_{perc}(NP = 500) < HV_{perc}(NP = 1000)$$

En la Figura 0.2 se muestra la zona de rechazo de tal prueba y el estadístico $z = -3.149$ del ejemplo, el cual también se localiza en zona de rechazo.

Figura 0.2 Ejemplo: gráfica de distribuciones normales (b)



De este modo podemos concluir de la tabla que, si el estadístico z es negativo y además la mitad de la probabilidad calculada ($\frac{0.0016}{2} = 0.0008$ en el ejemplo) es menor a α , entonces se puede rechazar la hipótesis H_0 de la prueba de cola izquierda y concluir que $HV_{perc}(NP = 1000)$ es estadísticamente mejor que $HV_{perc}(NP = 500)$. Lo contrario sucedería si el estadístico z fuera positivo (prueba de cola derecha).

Prueba de Kruskal-Wallis para k muestras independientes

Ésta consiste en probar la hipótesis nula

$$H_0: \mu_1 = \mu_2 = \dots = \mu_k$$

La hipótesis alternativa es general y establece únicamente que los datos no provienen de distribuciones idénticas. Las únicas condiciones para esta prueba son que se tengan mediciones ordinales y las distribuciones de las poblaciones sean continuas. El estadístico se encuentra aproximado por una distribución

chi-cuadrada con $k - 1$ grados de libertad. Así, la forma en la que se interpreta la información es la siguiente: si la probabilidad calculada es menor al nivel de significancia deseado, el estadístico está ubicado en zona de rechazo, de lo contrario es posible concluir que las muestras provienen de distribuciones distintas.

Se muestra un ejemplo en la Tabla 0.2, en el cual se desea probar si múltiples valores del parámetro θ en el MOEA/D muestran estadísticamente la misma calidad en las soluciones obtenidas de acuerdo a la métrica HV_{perc} , con un nivel de significancia de $\alpha = 0.05$. Dado que $0.1589 > \alpha$ no se puede rechazar la hipótesis nula y concluimos que no hay una diferencia en la calidad de los resultados al usar cualquiera de los 4 valores probados. En la Figura 0.3 se muestra la gráfica de distribución de la prueba.

Tabla 0.2 Ejemplo: Prueba de Kruskal-Wallis para k muestras independientes

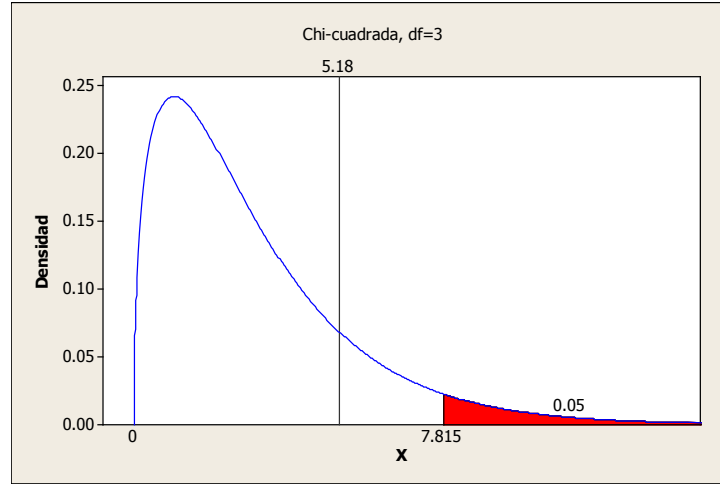
kruskal-wallis equality-of-populations rank test

theta	obs	Rank Sum
10	90	14464.00
30	90	16167.00
100	90	17444.00
1000	90	16905.00

chi-squared = 5.182 with 3 d.f.
probability = 0.1589

chi-squared with ties = 5.182 with 3 d.f.
probability = 0.1589

Figura 0.3 Ejemplo: gráfica de distribución chi-cuadrada con 3 grados de libertad



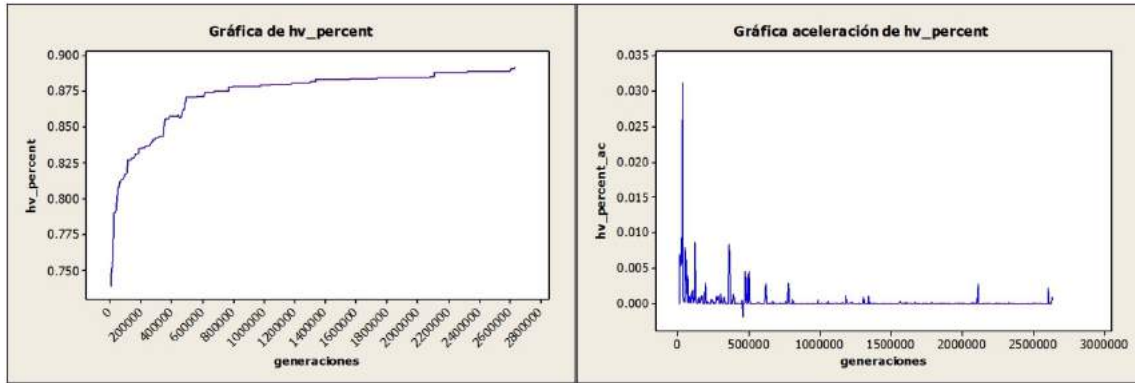
El problema del tamaño de archivo EP

Aún no se ha comentado la métrica cardinalidad, propuesta para evaluar las técnicas estudiadas en este proyecto. Retomando la Figura 3.9 queda claro que la cardinalidad de la población final en el MOEA/D es excesivamente más alta a comparación de las entregadas por NSGA-II. En la sección 3.4.1, se observó la imposibilidad de que MOEA/D alcanzara el 90% del hipervolumen real a partir de inst2. Se repitió el mismo experimento para esta instancia, pero dejando un límite de llamadas a la función objetivo muy alto (cercano a 80 millones), para ver qué nivel lograba alcanzar, el mejor valor obtenido fue $HV_{perc} = 89.1565\%$ ³⁶. En la Figura 0.4 del lado izquierdo se muestra cómo evoluciona el valor de la métrica HV_{perc} en el transcurso de esta ejecución. Para entender el lado derecho, es definido un factor nuevo nombrado aceleración de la métrica de la siguiente manera $\alpha = \frac{\Delta m}{\Delta G} = \frac{m_t - m_{t-1}}{G_t - G_{t-1}} = \frac{m_t - m_{t-1}}{1} = m_t - m_{t-1}$ donde m es

³⁶ Se utilizó la siguiente configuración de parámetros: tasa de mutación 0.01, $NP = 30$, $T = 4$, $\theta = 1.2$, el manejo de restricciones se hizo mediante el método max-min (revisar sección 3.6)

alguna métrica, así pues este factor reporta las variaciones observadas para la métrica m en cada nueva generación.

Figura 0.4 Ejecución exhaustiva inst2: HV_{perc}



De acuerdo a lo anterior se concluye que las mejoras más significativas del algoritmo ocurren en las primeras 800,000 generaciones y, posterior a esto, se presentan largos periodos (200,000 generaciones aproximadamente) sin mejora alguna. A todo esto, aunque de una forma sumamente lenta, siempre están mejorando las soluciones. A continuación se analiza la métrica de cardinalidad, que en este caso refiere al tamaño del archivo *EP*. De acuerdo a la Figura 0.5, el crecimiento del archivo *EP* es sostenido (izquierda) a pesar de que en la aceleración (derecha) muestra una alternancia constante de aumentos y decrementos en su tamaño. El archivo externo llega a alcanzar tamaños mayores a 120,000 elementos, lo cual es una cantidad exorbitante para un algoritmo que hace comparaciones de dominancia, es por lo tanto necesario limitar el tamaño de este archivo.

Figura 0.5 Ejecución exhaustiva inst2: cardinalidad

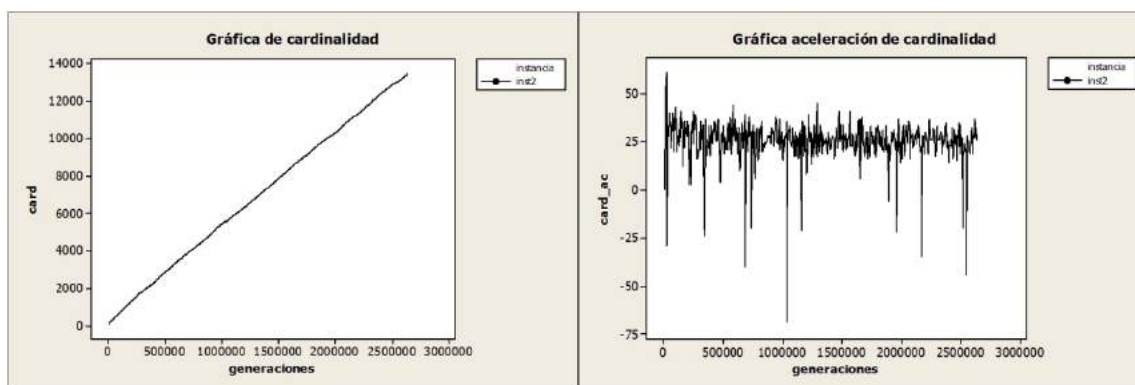
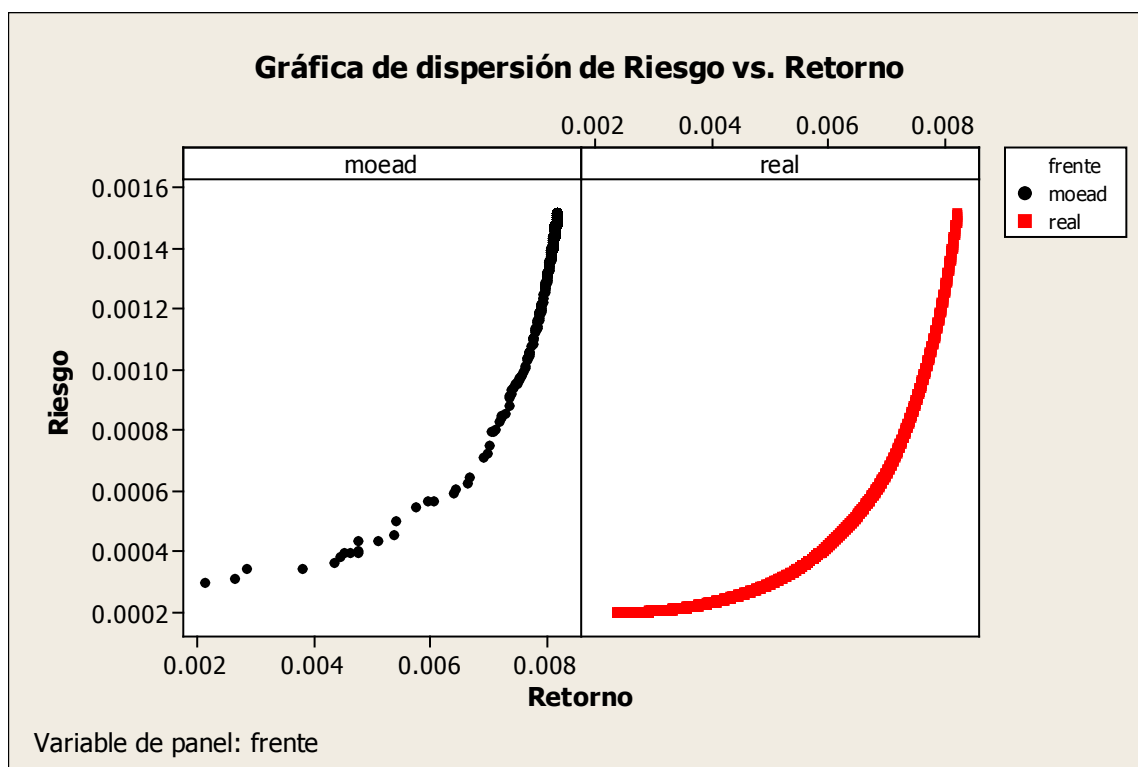


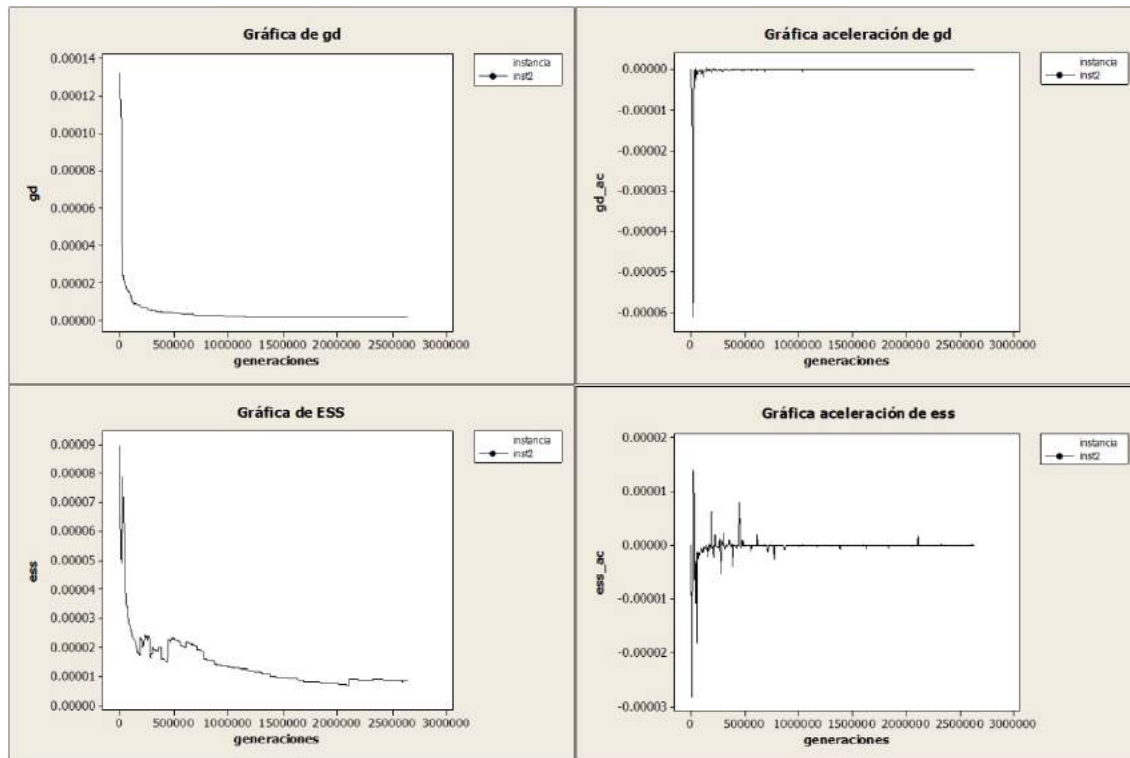
Figura 0.6 Ejecución exhaustiva inst2: Frente aproximado



Finalmente algo peculiar ilustrado en la Figura 0.6 es que la aproximación parece buena para valores más grandes de los objetivos, pero a medida que disminuyen, la aproximación a la curva parece ir desmejorando, ésta puede ser una característica propia del problema o de la estrategia. Las dos métricas

restantes se muestran en la Figura 0.7, las cuales tienen un comportamiento similar a HV_{perc} aunque la distancia generacional decrece de manera uniforme y ESS por otro lado muestra fluctuaciones más pronunciadas.

Figura 0.7 Ejecución exhaustiva inst2: gd y ESS



Recordando el modo operativo del MOEA/D, Se maneja un archivo de almacenamiento llamado EP , en el cuál se guardan las soluciones no dominadas encontradas en el transcurso de una ejecución (ver sección 2.5). En dicho archivo cuando se agrega una solución no dominada en cada generación, se desechan aquellas que se vuelven dominadas (ver Figura 0.5). Pero manejar un archivo cuyo tamaño puede tender al infinito representa un problema de eficiencia, ya que a medida que éste se vuelve más grande, las pruebas de dominancia se tornan extremadamente costosas (ver sección 2.1). Este

comportamiento constituye, obviamente, una desventaja en el marco de un estudio comparativo como el presente, en el que la eficiencia computacional es un criterio de gran importancia.

El mecanismo propuesto para limitar el tamaño de *EP* es el siguiente: cuando *EP* rebase su tamaño máximo permitido³⁷ se implementa un criterio basado en Crowding distance como se muestra en el Algoritmo 0.1, el cual se enfoca específicamente en la etapa del MOEA/D en la que se agregan individuos a *EP*. De este modo, se desechan soluciones en base a un criterio de dispersión, eliminando las que se encuentran más cercanas a sus vecinos tal como se explicó en la sección 2.4.2

Algoritmo 0.1 propuesta de restricción de archivo EP en la estrategia MOEA/D

```

...
revisar si hijo domina a algún elemento de EP
si ningún elemento de EP domina a hijo:
    se integra hijo a EP
    se eliminan los dominados por hijo en EP
si cardinalidad de EP ≥ NP:
    calcular Crowding distance para todos los elementos de EP
    ordenar de menor a mayor Crowding distance a EP
    mientras cardinalidad de EP ≥ NP:
        eliminar primer elemento de EP
  
```

³⁷ El tamaño máximo permitido aquí es *NP* (por similitud con el NSGA-II, que arroja su población corriente como evaluación del frente de Pareto), a menos que se especifique otro valor

Ajuste de parámetros para las estrategias VEGA, NSGA-II y MOEA/D

VEGA *Valores extremos*

Para esta estrategia, no se observó relación alguna entre los valores de los parámetros y el tiempo (Figura 0.8), por lo tanto tal factor no será relevante. Por otro lado, mientras mayor es el tamaño de la población las soluciones son de mejor calidad (Figura 0.9), por lo tanto, en esta etapa es elegido el valor más grande para tal parámetro (1000).

Figura 0.8 Ajuste parámetros etapa1: tiempo (VEGA)

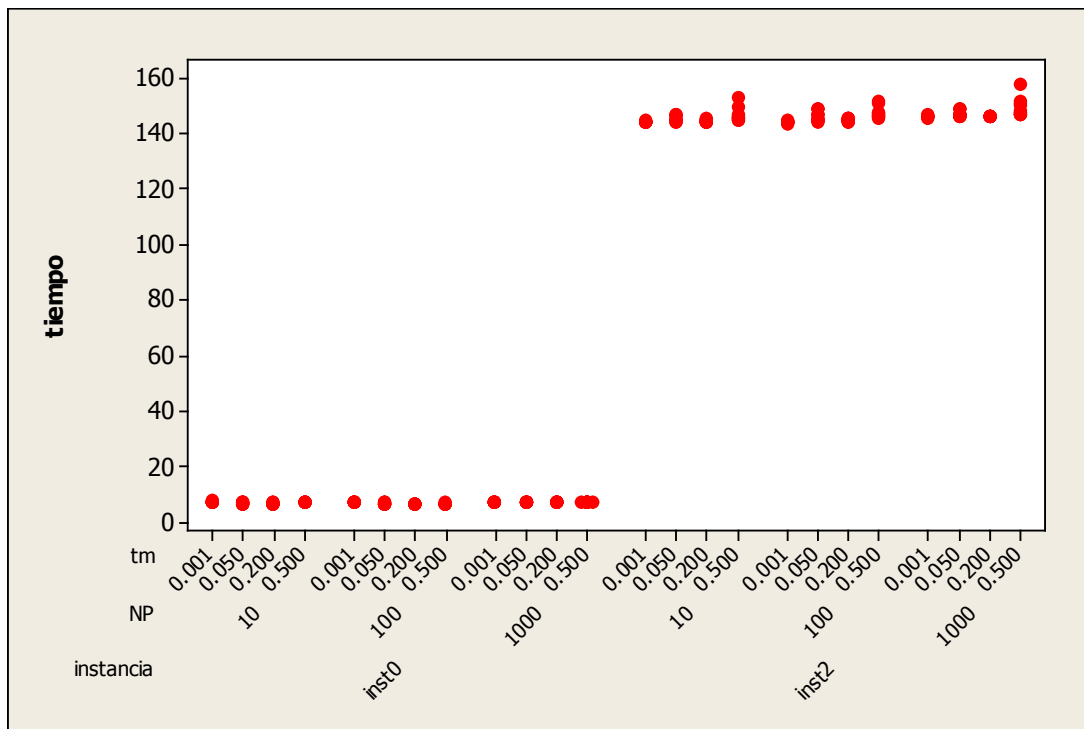
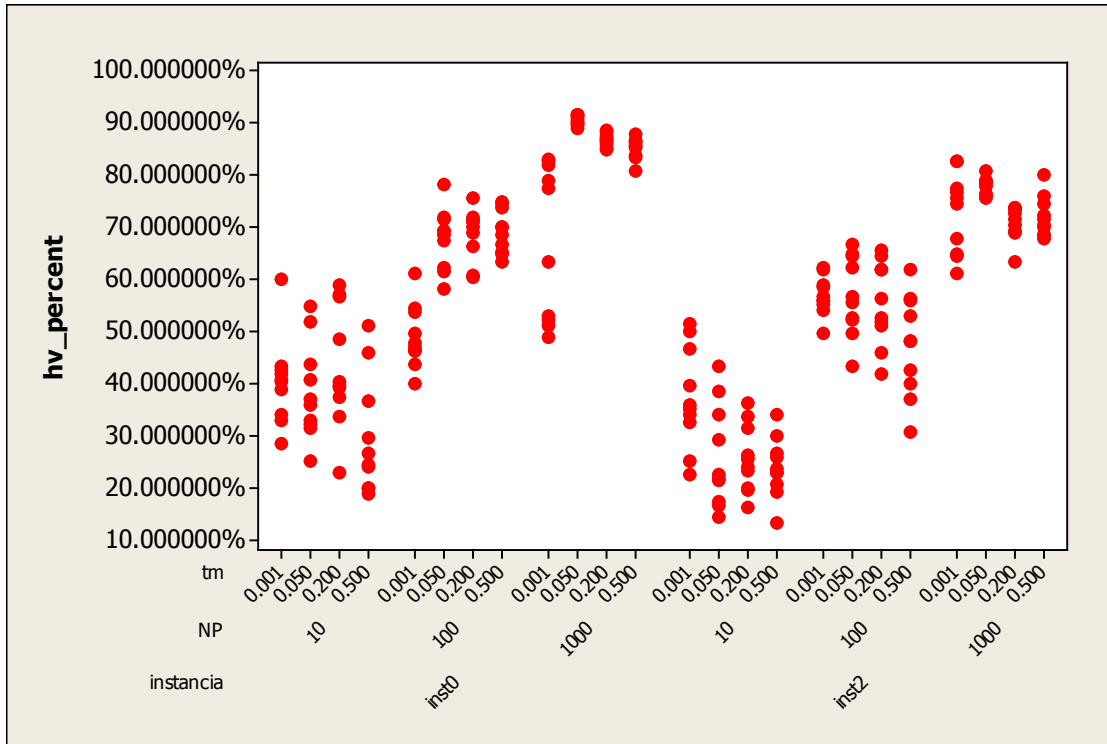


Figura 0.9 Ajuste parámetros etapa1: HV_{perc} (VEGA) (a)

De acuerdo a la decisión del tamaño de la población, la información se reduce a la Figura 0.10, que ilustra la misma tendencia que para la técnica anterior en la inst0 (mejores resultados para 0.05), pero en inst2 no es tan marcado tal efecto. Por lo anterior se tomará en cuenta el resto de las métricas (Figura 0.11) para elegir el valor adecuado de zm . Con respecto a distancia generacional, a medida que aumenta la tasa de mutación, la calidad de la soluciones empeora, así que es sensato elegir el menor valor posible de este parámetro. Pero volviendo a hipervolumen, en inst2 la varianza es muy alta para $zm = 0.001$, mientras que la más baja dispersión se obtiene con $zm = 0.05$ es la mejor. En base a los criterios anteriores se selecciona $zm = 0.05$.

Figura 0.10 Ajuste parámetros etapa1: HV_{perc} (VEGA) (b)

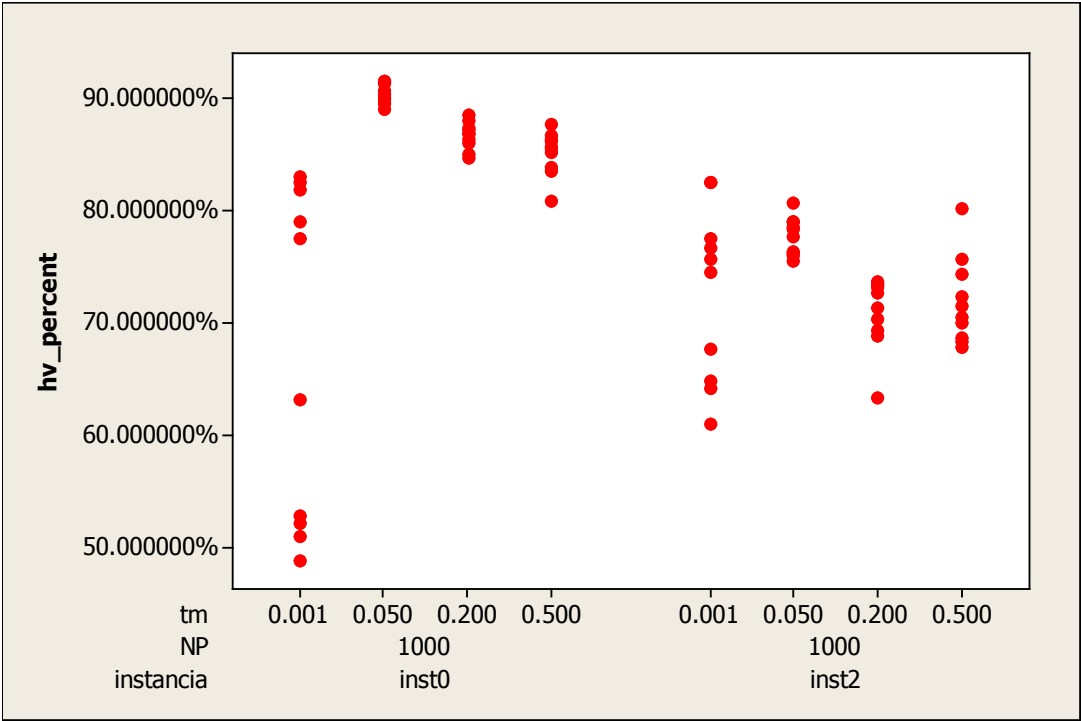
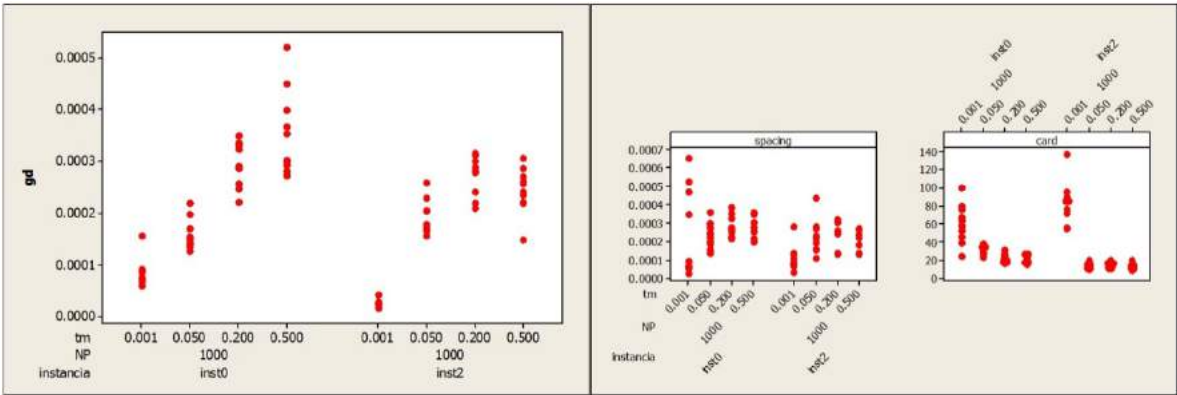


Figura 0.11 Ajuste parámetros etapa1: gd, ESS y card (VEGA)



Diseño factorial completo

Tabla 0.3 Valores de los parámetros etapa2 (VEGA)

Parámetro	Valores
Tasa de mutación (zm)	{0.01, 0.05 , 0.1}
Tamaño de la población	{20, 30, 500, 1000 }

Los valores probados en esta etapa se muestran en la Tabla 0.3. Ya se ha mencionado que el tiempo no es sensible al valor de los parámetros para la presente técnica, por lo cual se pasa directamente al análisis de las métricas.

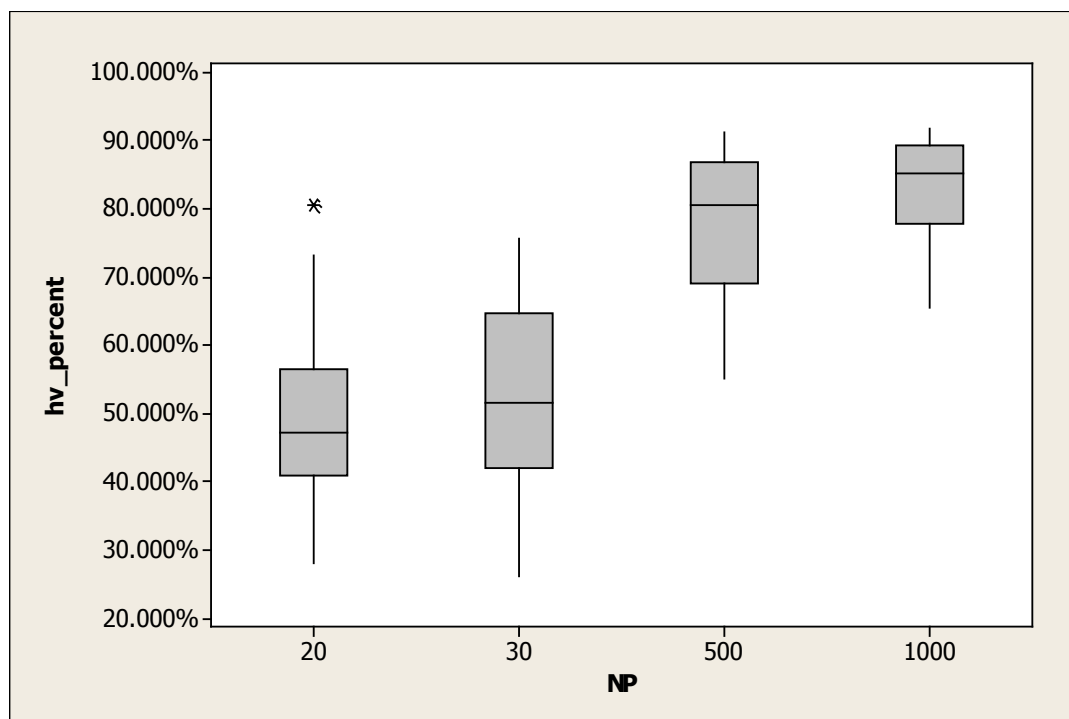
De acuerdo a la Figura 0.12 los mejores resultados se obtienen cuando $NP \geq 500$. En la Tabla 0.4 se muestra la prueba de una cola, donde $H_0: HV_{perc}(NP = 500) \geq HV_{perc}(NP = 1000)$, de acuerdo a los resultados se

Tabla 0.4 Ajuste parámetros etapa2: Prueba de Wilcoxon (VEGA) (a)

Two-sample wilcoxon rank-sum (Mann-whitney) test			
np	obs	rank sum	expected
500	60	3030	3630
1000	60	4230	3630
combined	120	7260	7260
unadjusted variance	36300.00		
adjustment for ties	0.00		
adjusted variance	36300.00		
Ho: hv_per~t(np==500) = hv_per~t(np==1000)			
z = -3.149			
Prob > z = 0.0016			

rechaza la hipótesis nula con $\alpha = 0.05$, por lo cual es elegido $NP = 1000$.

Figura 0.12 Ajuste parámetros etapa2: HV_{perc} (VEGA) (a)



Teniendo el parámetro tamaño de población fijo, la parte izquierda de la Figura 0.13 muestra los datos agrupados por tasas de mutación. Para cada instancia, parece ser que el mejor valor es diferente. Nuevamente, se realiza una prueba para el hipervolumen con respecto a tales tasas de mutación, presentada en la Tabla 0.5. Se concluye que no es posible rechazar la hipótesis nula de igualdad (prueba de dos colas) de HV_{perc} con $\alpha = 0.05$ por lo que no se puede justificar estadísticamente diferencias de eficiencia entre las dos tasas de mutación probadas. Por otro lado, en la misma Figura 0.13 del lado derecho, donde se muestra únicamente los datos para $zm = 0.01$, la dispersión de los datos es menor (más cercanos a la mediana), además de que al observar la distancia generacional en la Figura 0.14, tal parámetro alcanza los mejores valores para $zm = 0.01$, el cual se elige para la presente etapa.

Figura 0.13 Ajuste parámetros etapa2: HV_{perc} (VEGA) (b)

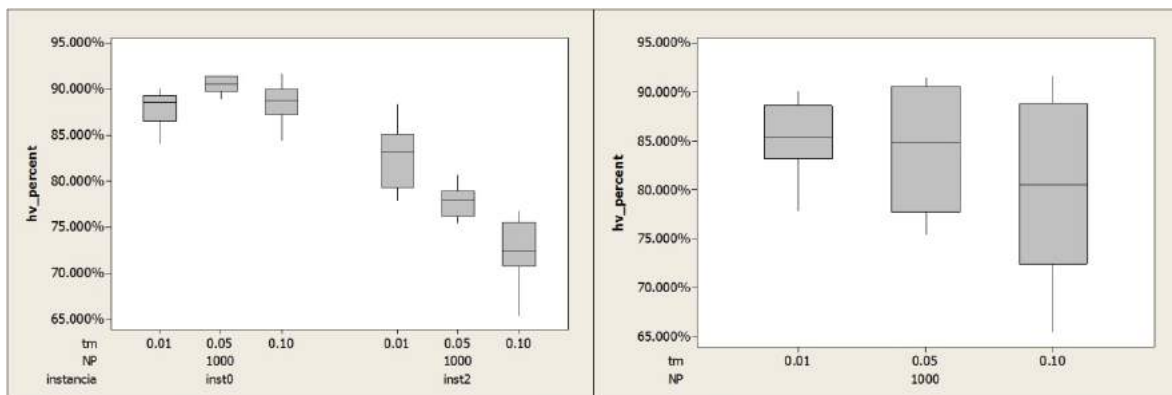
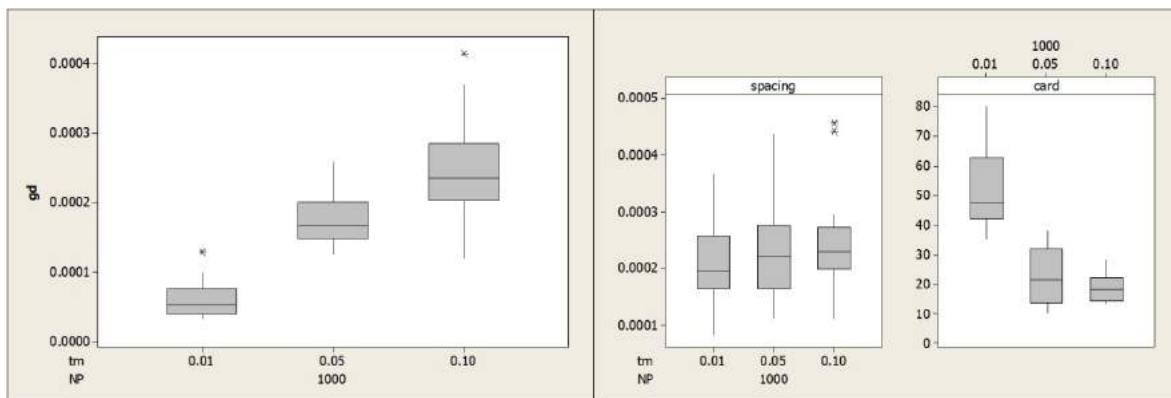


Tabla 0.5 Ajuste parámetros etapa2: Prueba de Wilcoxon (VEGA) (b)

Two-sample wilcoxon rank-sum (Mann-whitney) test			
tm	obs	rank sum	expected
.00999999977	20	405	410
.05000000074	20	415	410
combined	40	820	820
unadjusted variance	1366.67		
adjustment for ties	0.00		
adjusted variance	1366.67		
Ho: hv_per~t(tm==.0099999997764826) = hv_per~t(tm==.0500000007450581)			
z = -0.135			
Prob > z = 0.8924			

Figura 0.14 Ajuste parámetros etapa2: gd, ESS y card (VEGA)



NSGA-II

Valores extremos

En esta etapa de la investigación es forzoso no perder de vista el efecto de las pruebas de dominancia en el tiempo de ejecución de los algoritmos. En la técnica SPO (sección 3.7.1), ya fue relevante tomar decisiones basados en el criterio tiempo, aun tratándose de una técnica donde las pruebas de dominancia se utilizan únicamente en la población final. Pero el modo operativo del NSGA-II en cambio se basa en el paradigma de dominancia de Pareto y la eficiencia de los métodos que realizan tales pruebas se relaciona directamente con el tamaño de las poblaciones experimentales (sección 2.1).

Dado que la presente es una etapa de valores extremos, ya de antemano es sabido que sería una mala idea acercarse al extremo superior del intervalo para el parámetro tamaño de población. De hecho, se nota el efecto drástico de la variable temporal, tan solo para la instancia más pequeña en la Figura 0.15. Mientras que los tiempos para tamaños de población de 10 y 100 oscilan entre 7.74 y 31.33, segundos para 1000 se alcanzan valores de hasta 1923 segundos (32 min) y de hecho los resultados obtenidos usando poblaciones de 1000 individuos tampoco son significativamente superiores en cuanto a calidad con respecto a los experimentos con $NP = 100$ (Figura 0.16).

Pero por otro lado, puede haber una zona de valores de NP donde es posible manejar este parámetro de manera eficiente, como fue discutido en la sección 2.1.4: respecto a las funciones de complejidad (Figura 2.2) de los algoritmos de esa sección, para valores pequeños los tiempos pueden ser muy buenos.

Figura 0.15 Ajuste parámetros etapa1: tiempo (NSGA-II)

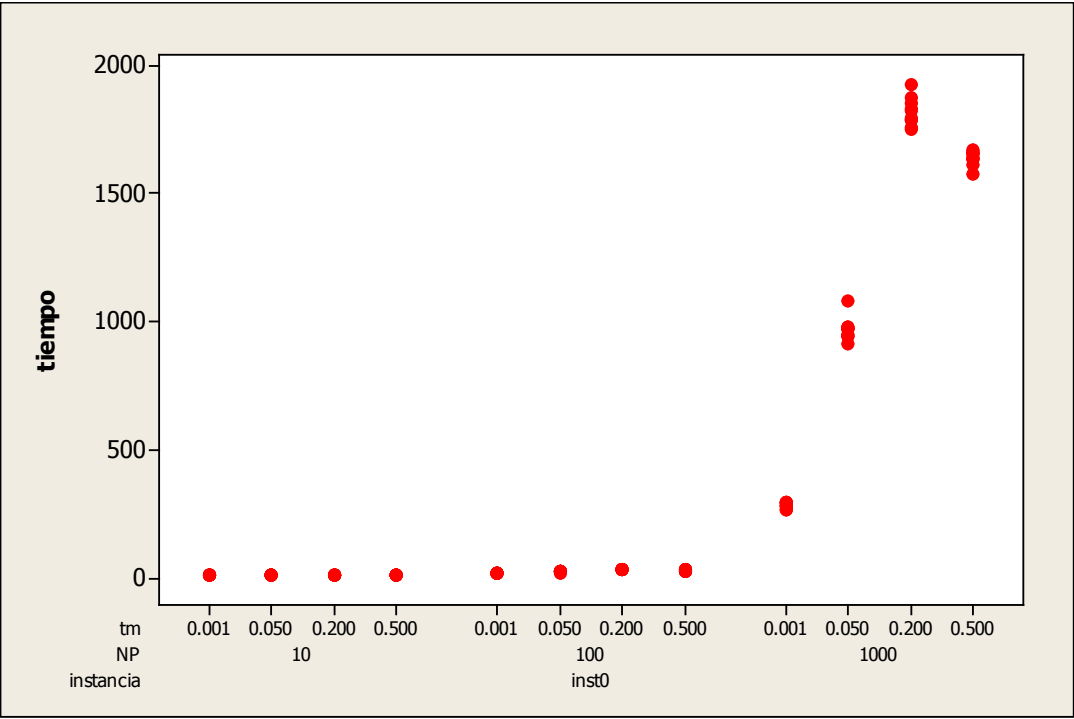
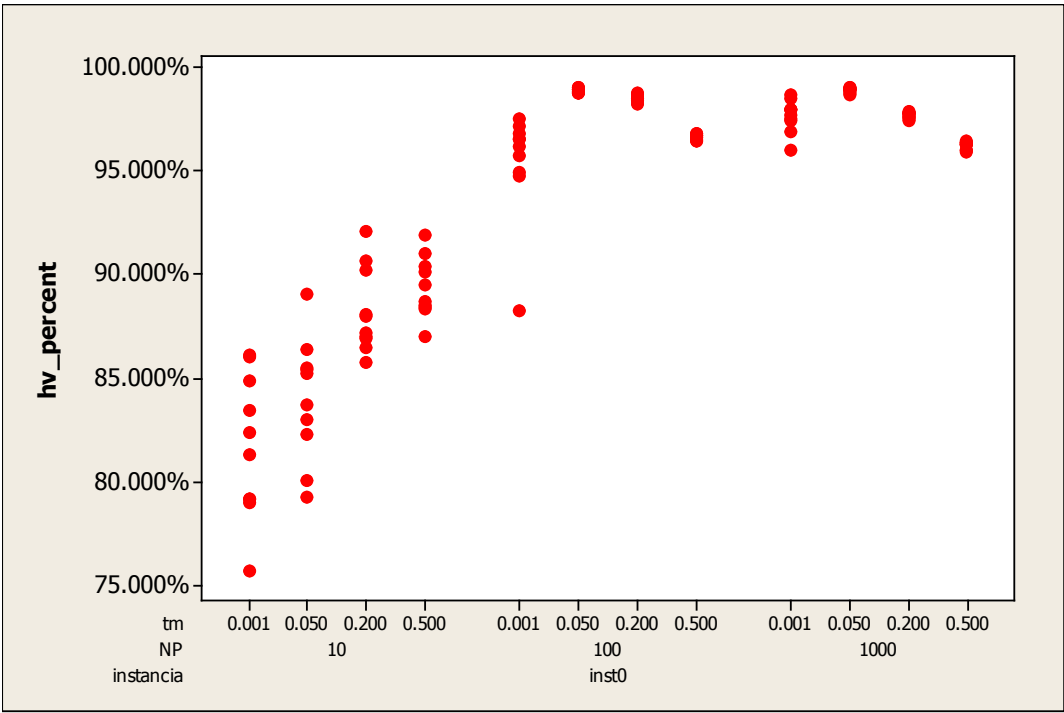


Figura 0.16 Ajuste parámetros etapa1: HV_{perc} (NSGA-II) (a)



Por las razones anteriores, hemos descartado de antemano el valor más grande. De esta manera los datos quedan como en la Figura 0.17 y de inmediato, es claro que para ambas instancias, los mejores resultados son para $NP = 100$. De acuerdo a la prueba de Wilcoxon mostrada en la Tabla 0.6, la Hipótesis nula de igualdad de hipervolúmenes para $NP = 10$ contra $NP = 100$ se rechaza con un nivel de significancia de 0.05, es seleccionado de este modo $NP = 100$.

Figura 0.17 Ajuste parámetros etapa1: HV_{perc} (NSGA-II) (b)

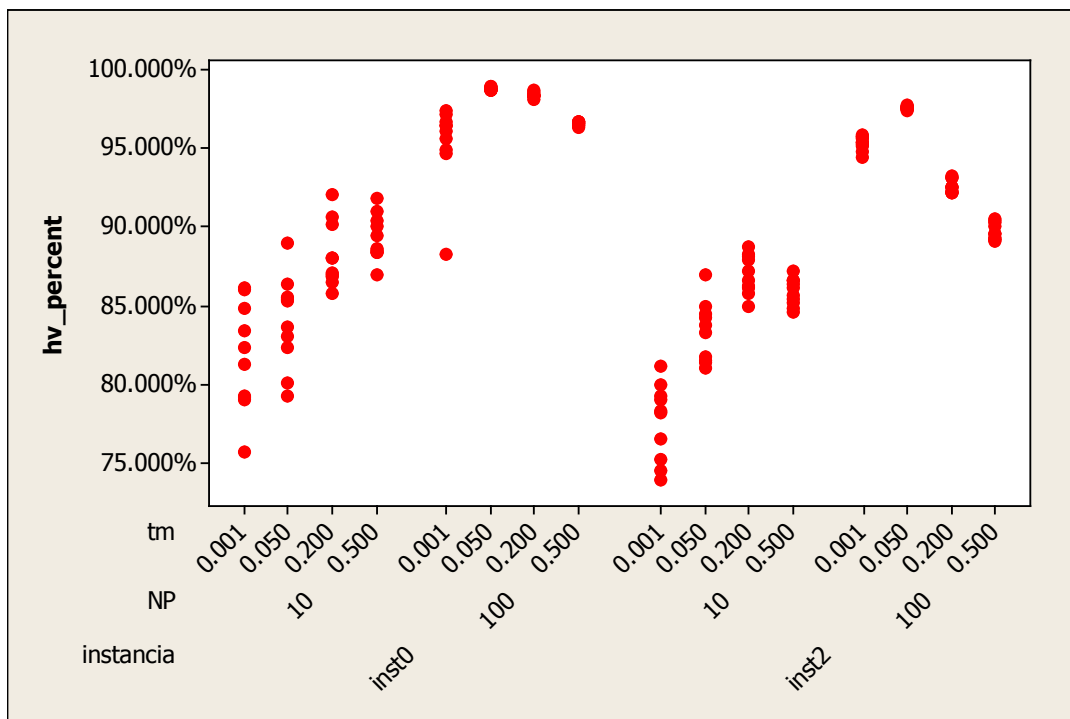
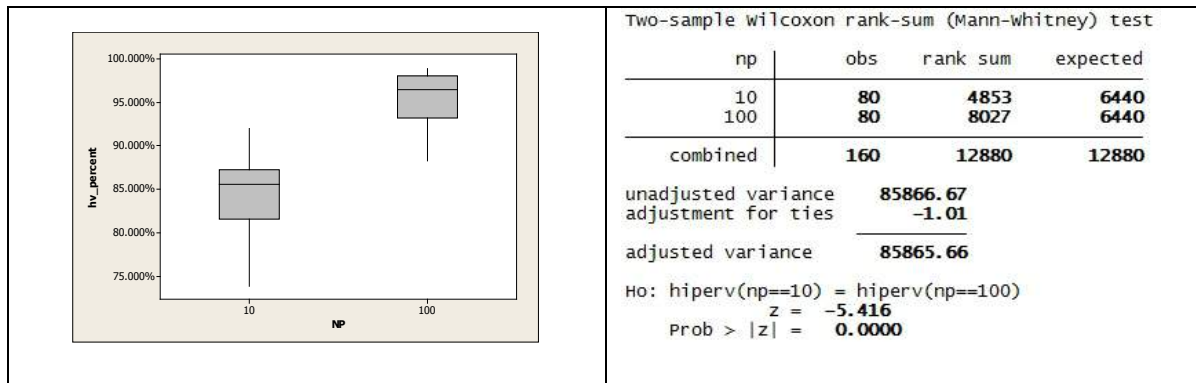
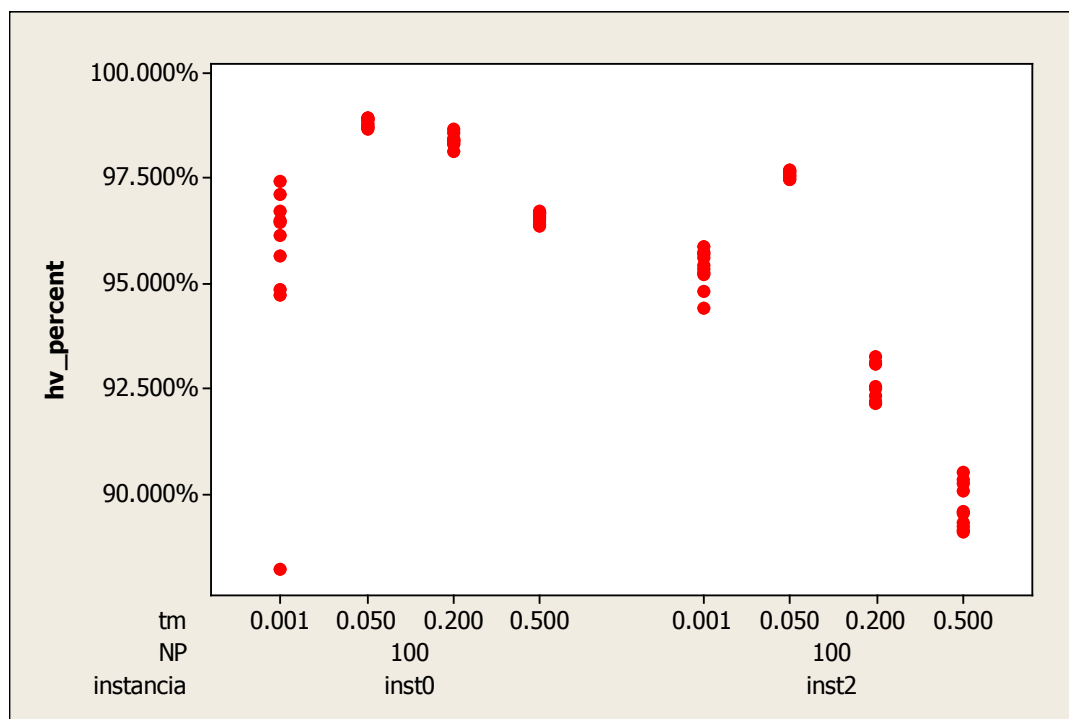
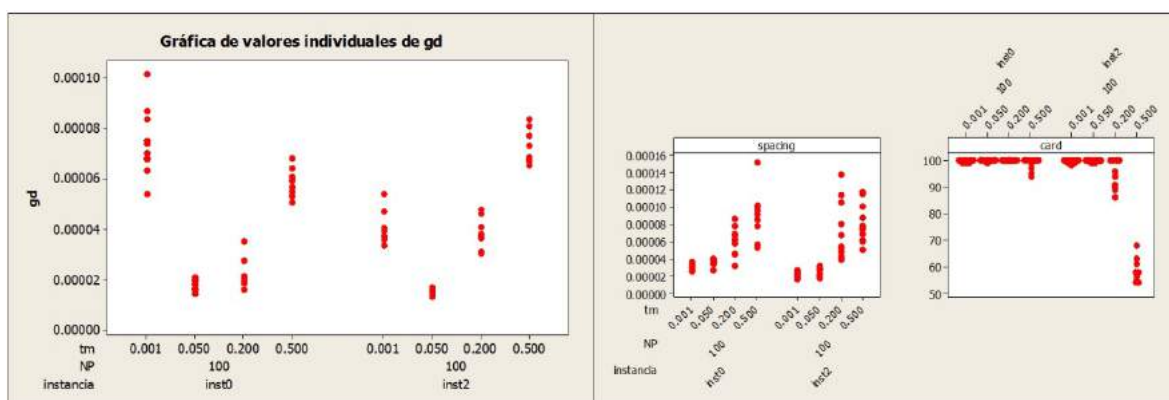


Tabla 0.6 Ajuste parámetros etapa1: Prueba de rangos de signos de Wilcoxon (NSGA-II)



Elegido el tamaño de la población adecuado, solo queda por decidir la tasa de mutación. Es sencillo notar que para inst2 sería $zm = 0.05$ (Figura 0.18), por otro lado se observa que al aumentar la tasa de mutación la solución mejora hasta un punto ($zm = 0.05$) donde alcanza sus máximos valores para HV_{perc} y a partir de ahí se deteriora la calidad de la solución. El efecto inverso se produce para gd de acuerdo a la Figura 0.19 de este modo se elige 0.05 como el valor para tal parámetro.

Figura 0.18 Ajuste parámetros etapa1: HV_{perc} (NSGA-II) (c)Figura 0.19 Ajuste parámetros etapa1: gd , ESS y $card$ (NSGA-II)

Observe que, para tasas de mutaciones altas las cardinalidades algunas veces son menores (Figura 0.19), pero son éstas las que logran mejores soluciones en cuanto a convergencia y dispersión.

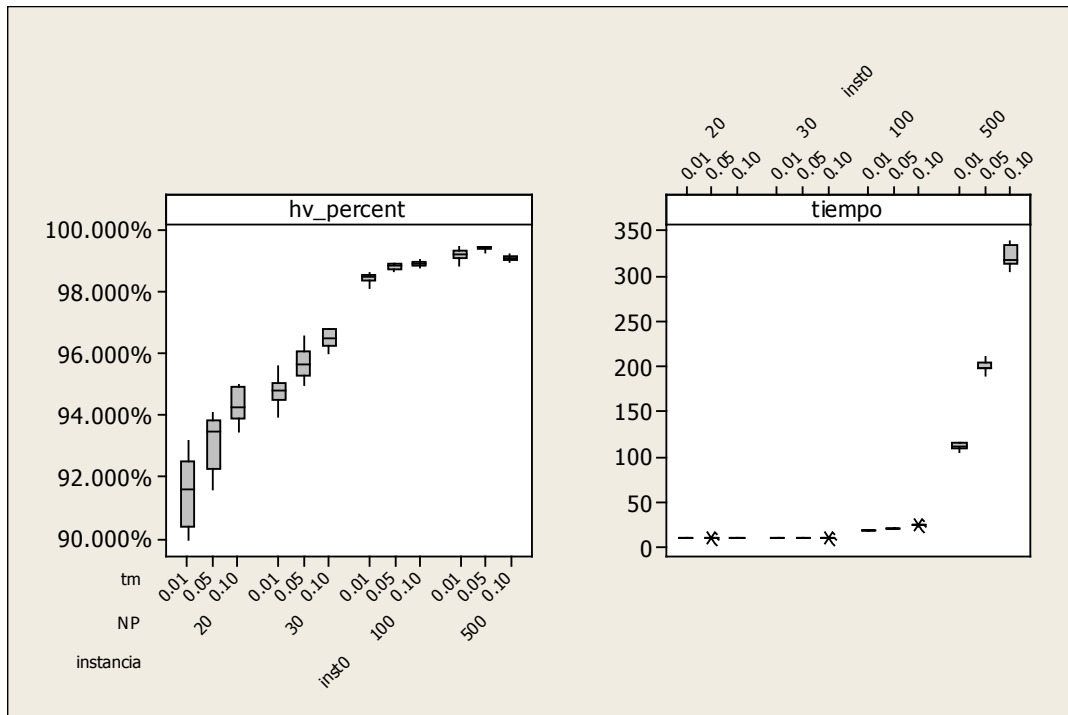
Diseño factorial completo

La Figura 0.20 aún presenta una mejora con manejo de poblaciones grandes, sin embargo el costo temporal es muy significativo para $NP = 500$ en comparación de la mejora de calidad al saltar de 100 a 500. Por tanto éste último valor es descartado.

Tabla 0.7 Valores de los parámetros etapa2 (NSGA-II)

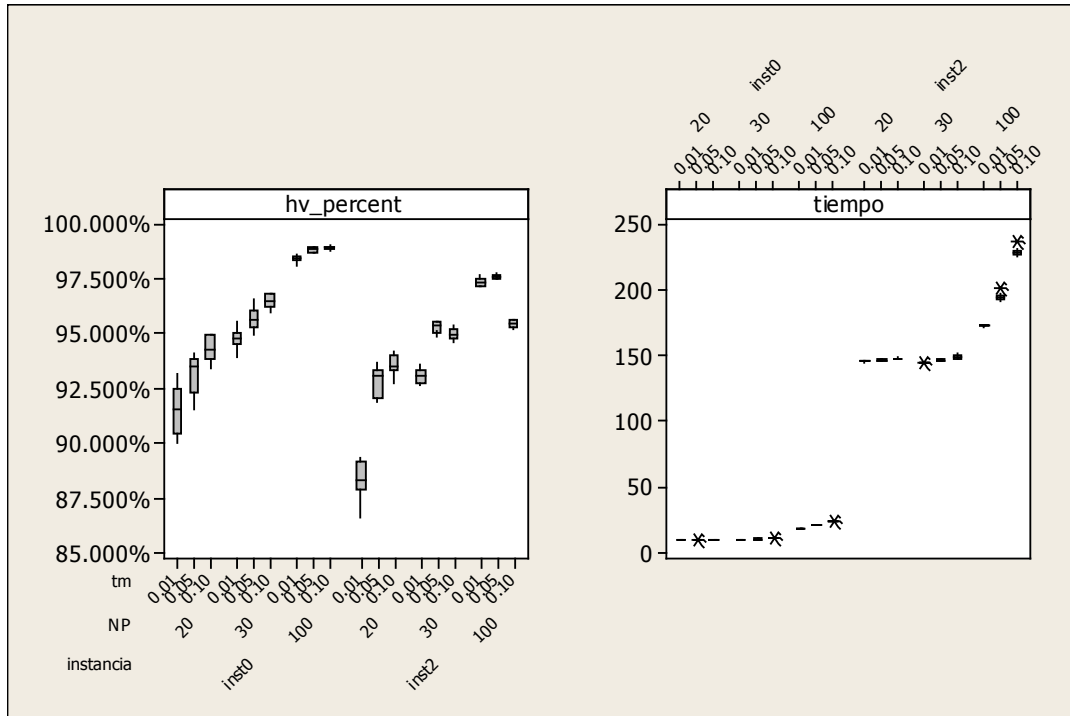
Parámetro	Valores
Tasa de mutación (zm)	{0.01, 0.05 , 0.1}
Tamaño de la población	{20, 30, 100 , 500}

Figura 0.20 Ajuste parámetros etapa2: HV_{perc} y tiempo (NSGA-II) (a)



Por otro lado el efecto $NP - tiempo$ sigue siendo notable (Figura 0.21) en ambas instancias, la cuestión es si es significativo.

Figura 0.21 Ajuste parámetros etapa2: HV_{perc} y tiempo (NSGA-II) (b)



Volviendo a la calibración de las estrategias anteriores. En la Figura 0.22 se muestran los tiempos para las anteriores dos técnicas con los parámetros calibrados, además de los tiempos conseguidos por los diferentes tamaños de población en el NSGA-II; así como los estadísticos descriptivos correspondientes en la Tabla 0.8. Los tiempos de las anteriores técnicas son muy similares entre sí, así como también NSGA-II con poblaciones de 20 y 30, sin embargo para el tamaño 100, el tiempo ya supera significativamente los de las anteriores técnicas (para inst0 al menos un 147% mayor y para inst2 al menos 25% con respecto a la media). Se espera que tal efecto en el tiempo sea más marcado para instancias de mayor tamaño. Por tal motivo pareció conveniente, en cuanto a comparación de técnicas, utilizar el tamaño de $NP = 30$ ya que los

resultados son aceptables y fijado el criterio de paro los tiempos de ejecución son más homogéneos.

Figura 0.22 Ajuste parámetros etapa2: tiempo (NSGA-II)

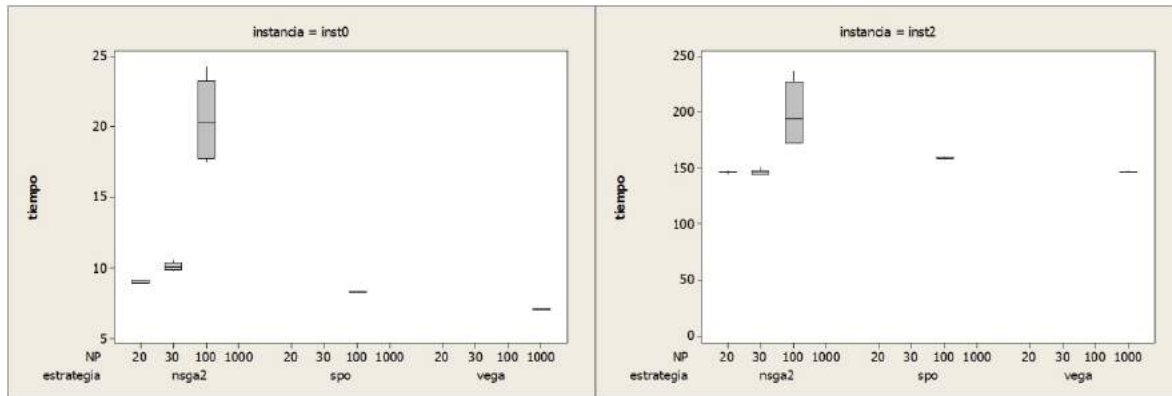


Tabla 0.8 Ajuste parámetros etapa2: tiempo (NSGA-II comparado con SPO y VEGA)

Instancia	Estrategia	Ψ	Media	Desv. Est.	Mediana
inst0	nsga2	20	8.98	0.12	8.94
inst0	nsga2	30	10.09	0.22	10.07
inst0	nsga2	100	20.48	2.40	20.30
inst0	spo	100	8.29	0.03	8.29
inst0	vega	1000	7.08	0.01	7.07
inst2	nsga2	20	146.39	0.88	146.52
inst2	nsga2	30	146.56	2.15	146.49
inst2	nsga2	100	198.74	23.72	193.95
inst2	spo	100	159.23	0.93	159.13
inst2	vega	1000	146.62	0.45	146.53

Con respecto a la tasa de mutación, la información está resumida en la Figura 0.23. Para inst0 la mejor opción es $zm = 0.1$, mientras que para inst2 es la tasa de 0.05. En base a la prueba de la Tabla 0.9, con un nivel de significancia de 0.05, no es posible rechazar la hipótesis de igualdad de hipervolumen para tasas de 0.05 y 0.1. Por otro lado, tampoco se puede rechazar la hipótesis para gd (Tabla 0.10), no queda más que hacer la elección respecto a otro criterio. Para

las métricas gd y ESS la tendencia es de mejora a medida que la tasa de mutación aumenta y de acuerdo a la Figura 0.24, la mejor mediana es para $zm = 0.1$, éste es el valor que se eligió para tal parámetro.

Figura 0.23 Ajuste parámetros etapa2: HV_{perc} (NSGA-II)

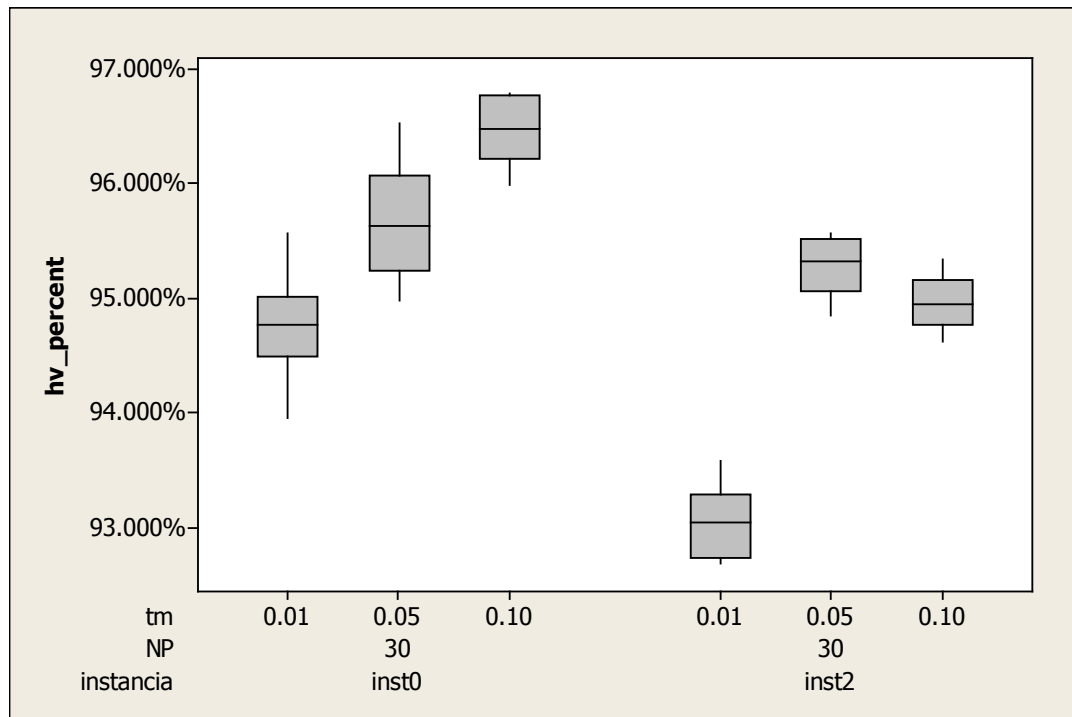
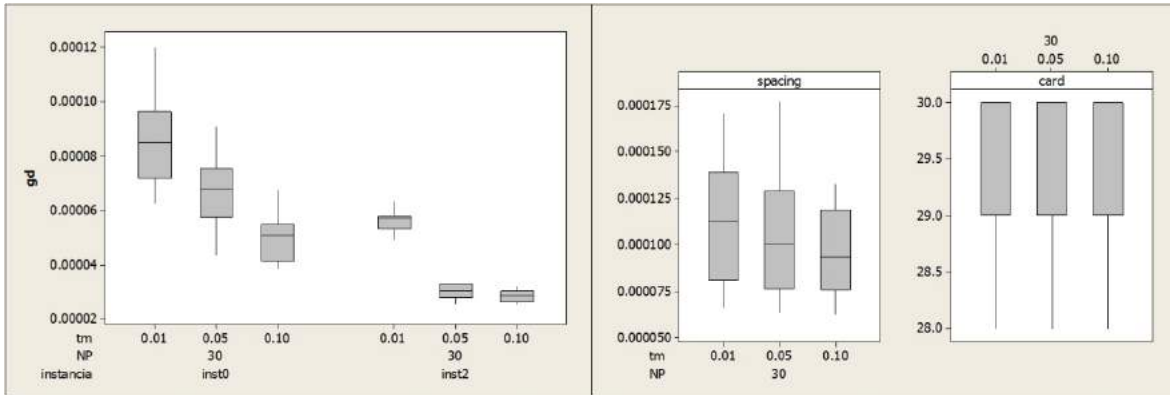


Tabla 0.9 Ajuste parámetros etapa2: Prueba de Wilcoxon HV_{perc} (NSGA-II)

Two-sample wilcoxon rank-sum (Mann-whitney) test			
tm	obs	rank sum	expected
.05000000074	20	394	410
.10000000149	20	426	410
combined	40	820	820
unadjusted variance	1366.67		
adjustment for ties	0.00		
adjusted variance	1366.67		
Ho: hv_per~t(tm==.0500000007450581) = hv_per~t(tm==.1000000014901161)			
z = -0.433			
Prob > z = 0.6652			

Figura 0.24 Ajuste parámetros etapa2: gd, ESS y card (NSGA-II)

Tabla 0.10 Ajuste parámetros etapa2: Prueba de Wilcoxon, *gd* (NSGA-II)

Two-sample wilcoxon rank-sum (Mann-whitney) test			
tm	obs	rank sum	expected
.050000000074	20	465	410
.100000000149	20	355	410
combined	40	820	820
unadjusted variance	1366.67		
adjustment for ties	0.00		
adjusted variance	1366.67		
Ho: $gd(tm=.05000000007450581) = gd(tm=.10000000014901161)$			
z =		1.488	
Prob > z =		0.1368	

MOEA/D Valores extremos

Esta estrategia tiene más parámetros que calibrar (Tabla 0.11). Dado que para un diseño factorial completo, el aumento de experimentos es muy grande cada vez que se agrega un parámetro (sección 3.3), se realiza únicamente el ajuste usando *inst0*. Los parámetros particulares de la presente estrategia son: el parámetro θ de Boundary Intersection Approach (ver sección 2.5.1) y el tamaño de vecindario (T).

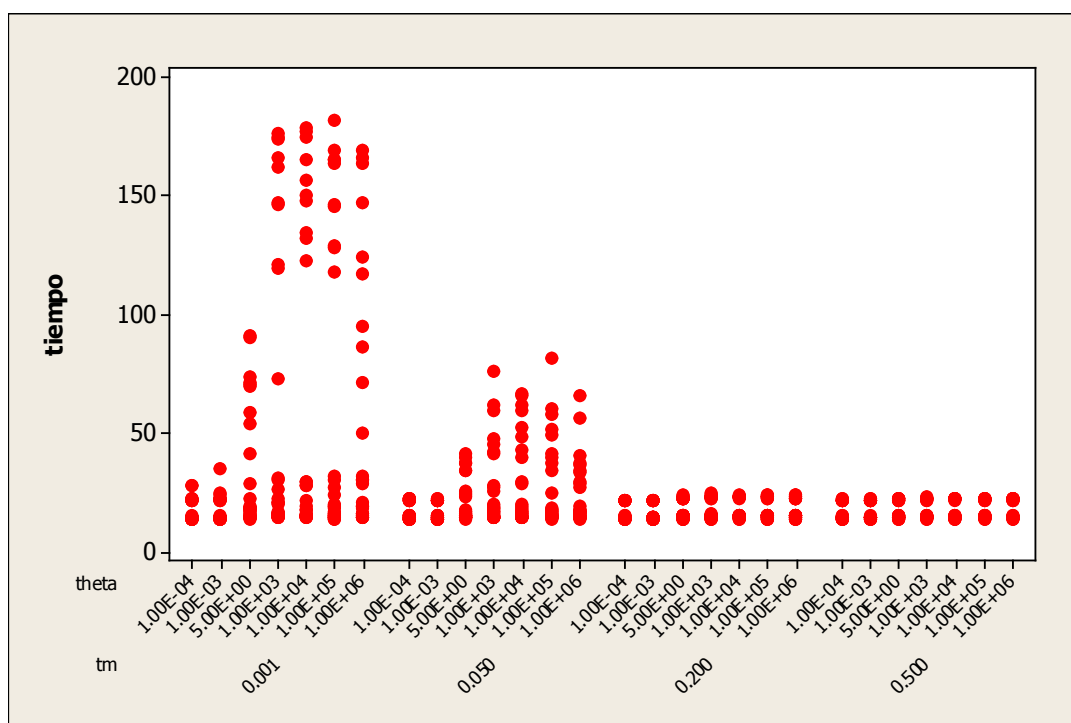
El tamaño del vecindario es el número de soluciones con las que cada individuo compartirá información, no se espera el mismo efecto de un vecindario tamaño 4 para una población de 20 que para una de 100, por lo tanto se calibra el porcentaje de la población total que representa el tamaño del vecindario. Este último se realizó en una sola etapa dado que los valores son pocos.

Tabla 0.11 Valores de los parámetros etapa1 (MOEA/D)

Parámetro	Valores
Tasa de mutación (zm)	{0.001, 0.05, 0.2, 0.5}
Tamaño de la población	{10, 100, 1000}
Theta (θ)	{0.0001, 0.001, 5, 1000, 10000, 100000, 1000000}
Tamaño del vecindario en porcentaje (T)	{10%, 13.33%, 20%}

La primera información relevante que se da a lucir: tasas de mutación más altas permiten tener tiempos más eficientes (Figura 0.25). Por otro lado, al analizar esta variable a detalle, como era de esperarse, el manejo de poblaciones muy grandes provoca que se disparen los tiempos de ejecución (Figura 0.26). Sin embargo es posible de acuerdo al gráfico que esto solo suceda bajo ciertas circunstancias que involucran otros parámetros.

Figura 0.25 Ajuste parámetros etapa1: tiempo (MOEA/D) (a)



Al abordar la métrica HV_{perc} (Figura 0.27) es visible, de igual forma, que para una tasa de mutación muy baja se han obtenido los peores resultados. Se inicia entonces por descartar tal valor (0.001). Por otro lado, si bien los valores restantes muestran eficiencias similares en el mismo gráfico, regresando al tiempo (Figura 0.25) claramente éste es más estable para tasas mayores o iguales a 0.2 (la dispersión es menor), además de que nunca alcanzan tiempos mayores al promedio $(25.356)^{38}$. De esta forma se decide que la tasa de mutación debe ser elegida entre 0.2 y 0.5 para esta etapa.

³⁸ Los máximos tiempos alcanzados para las tasas de mutación 0.2 y 0.5 son 25.2 y 23.61 segundos respectivamente.

Figura 0.26 Ajuste parámetros etapa1: tiempo (MOEA/D) (b)

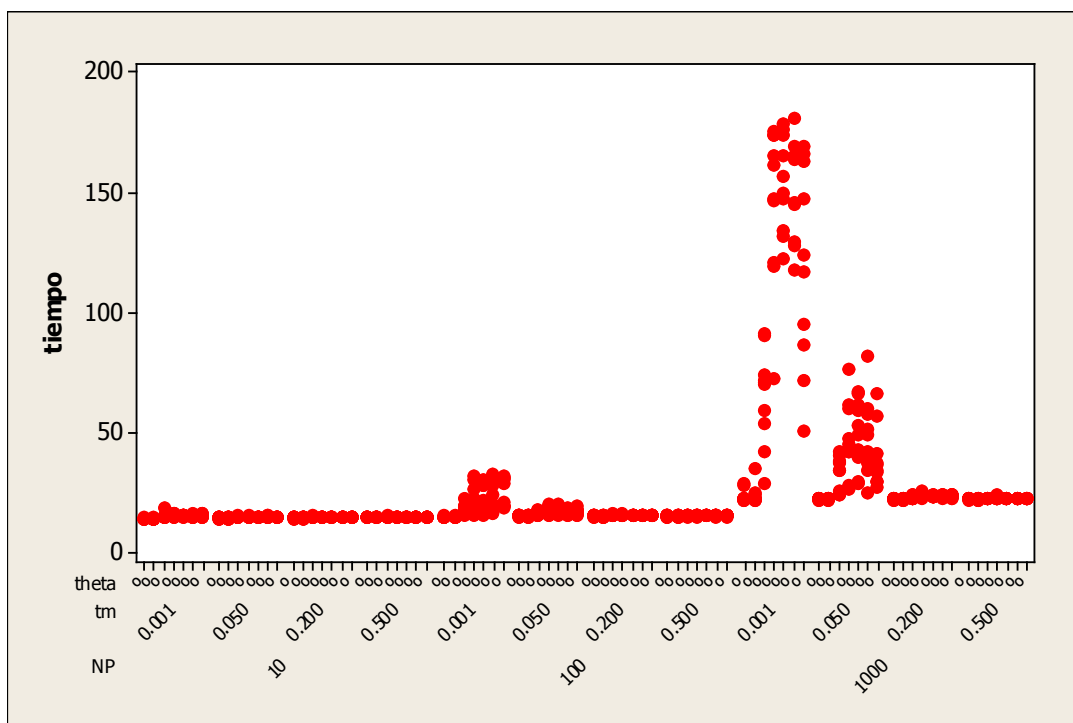
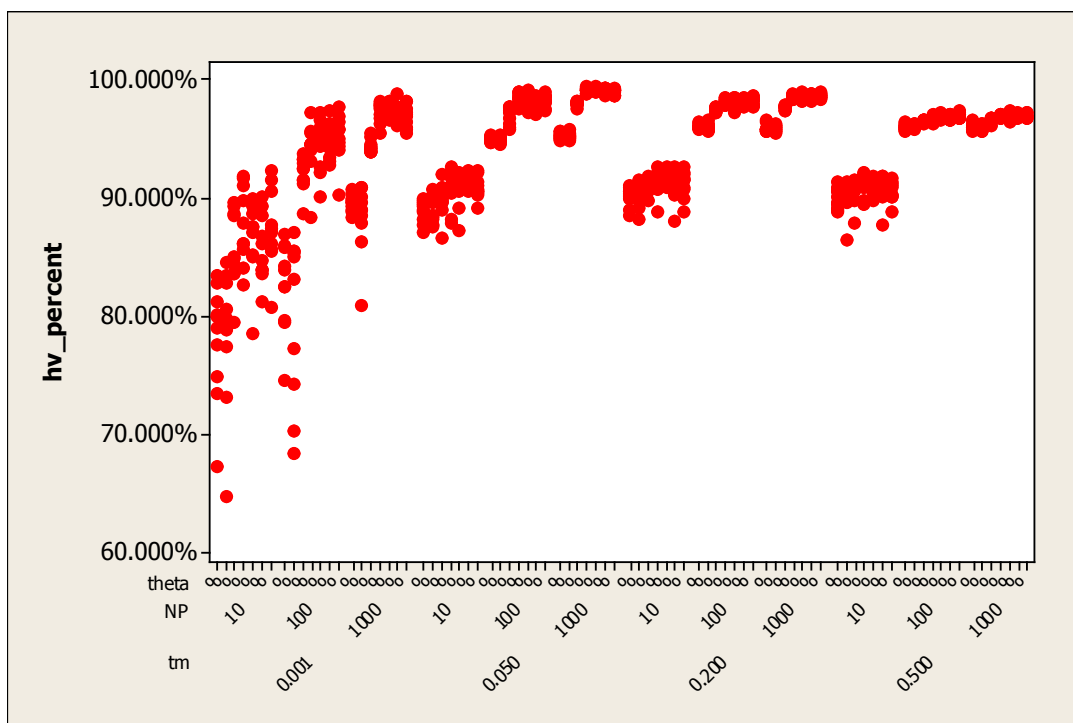
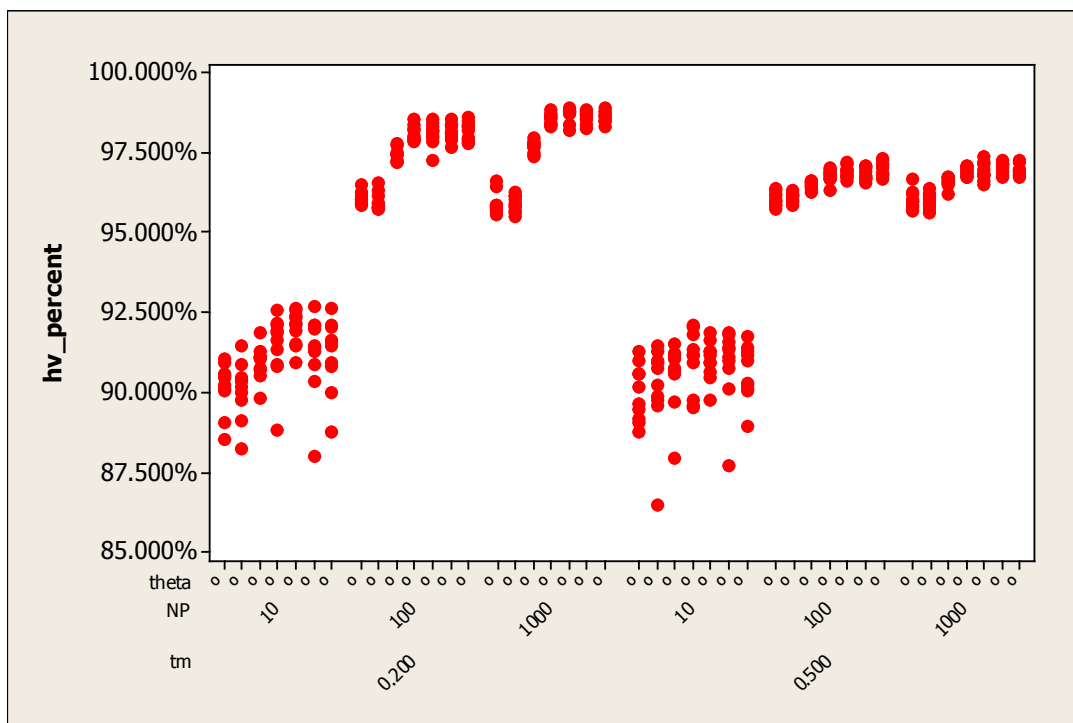


Figura 0.27 Ajuste parámetros etapa1: HV_{perc} (MOEA/D) (a)



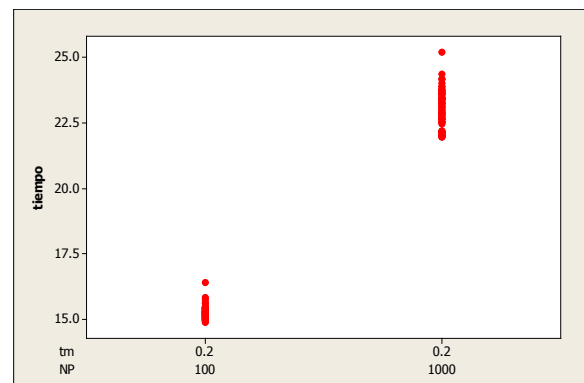
De acuerdo a lo anterior se tiene la información como se muestra en la Figura 0.28. Para las tasas restantes, el comportamiento parece ser similar al variar el tamaño de la población, pero, para múltiples valores de theta, definitivamente son superiores los resultados obtenidos con $zm = 0.2$ el cual es el valor seleccionado en la presente etapa.

Figura 0.28 Ajuste parámetros etapa1: HV_{perc} (MOEA/D) (b)



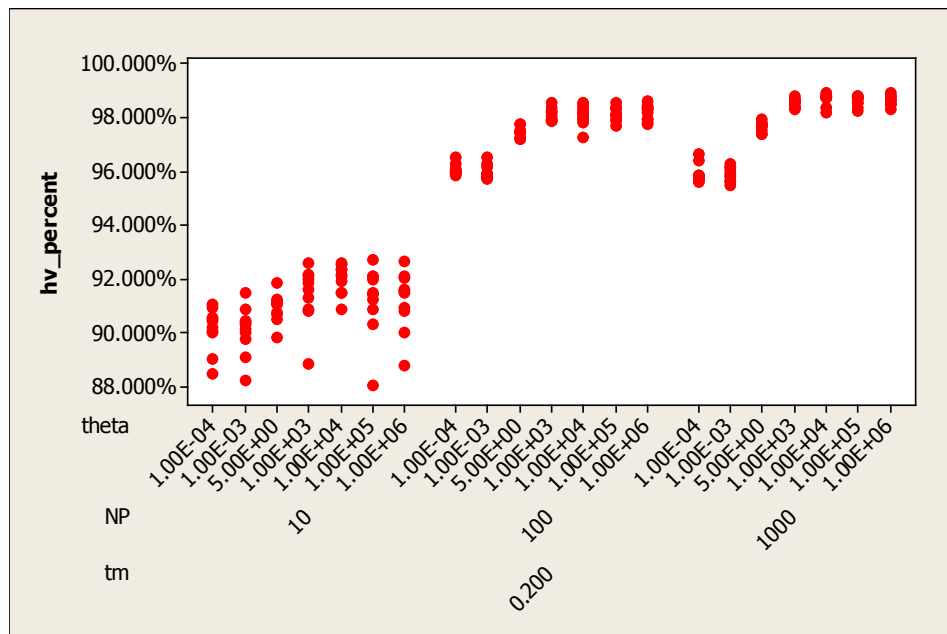
Ya teniendo un valor fijo para zm se puede analizar de manera más clara los resultados. En la Figura 0.30, ya no hay una forma visible de elegir algún valor para el tamaño de la población entre 100 y 1000. Se analizó antes la relevancia

Figura 3.29 Ajuste parámetros etapa1: tiempo (MOEA/D) (c)



del manejo de poblaciones pequeñas para el NSGA-II. En el MOEA/D también, el manejo del archivo EP exige hacer considerables pruebas de dominancia, por tanto es conveniente manejar poblaciones lo más pequeñas que sea posible, sin alterar la eficiencia del algoritmo. Note en la Figura 3.60 que para el valor seleccionado de zm los tiempos son considerablemente mejores para $NP = 100$.

Figura 0.30 Ajuste parámetros etapa1: HV_{perc} (MOEA/D) (c)



Solo falta por elegir el valor adecuado para θ , teniendo fijos los valores de las demás métricas. En la Figura 0.31, para los valores probados, se puede apreciar que a partir de $\theta = 1000$ se alcanzan los

Tabla 0.12 Ajuste parámetros etapa1: prueba de Kruskal-Wallis (MOEA/D)

```
. kwallis hiperv if np==100 & theta >= 1000, by(theta)
```

Kruskal-wallis equality-of-populations rank test

theta	obs	Rank Sum
1000	10	183.00
10000	10	206.00
100000	10	212.00
1.0e+06	10	219.00

chi-squared = 0.534 with 3 d.f.
probability = 0.9113

chi-squared with ties = 0.534 with 3 d.f.
probability = 0.9113

mejores valores para el hipervolumen.

Dado que la prueba de Kruskal-Wallis para muestras independientes (Tabla 0.12) nos dice que no hay manera de rechazar con un nivel de significancia de 0.05 la hipótesis nula de igualdad de hipervolumenes para los valores mencionados, se elige 1000 dado que hay indicios de que valores grandes pueden disparar el tiempo. El valor para el resto de las métricas se muestra en la Figura 0.32.

Figura 0.31 Ajuste parámetros etapa1: HV_{perc} (MOEA/D) (d)

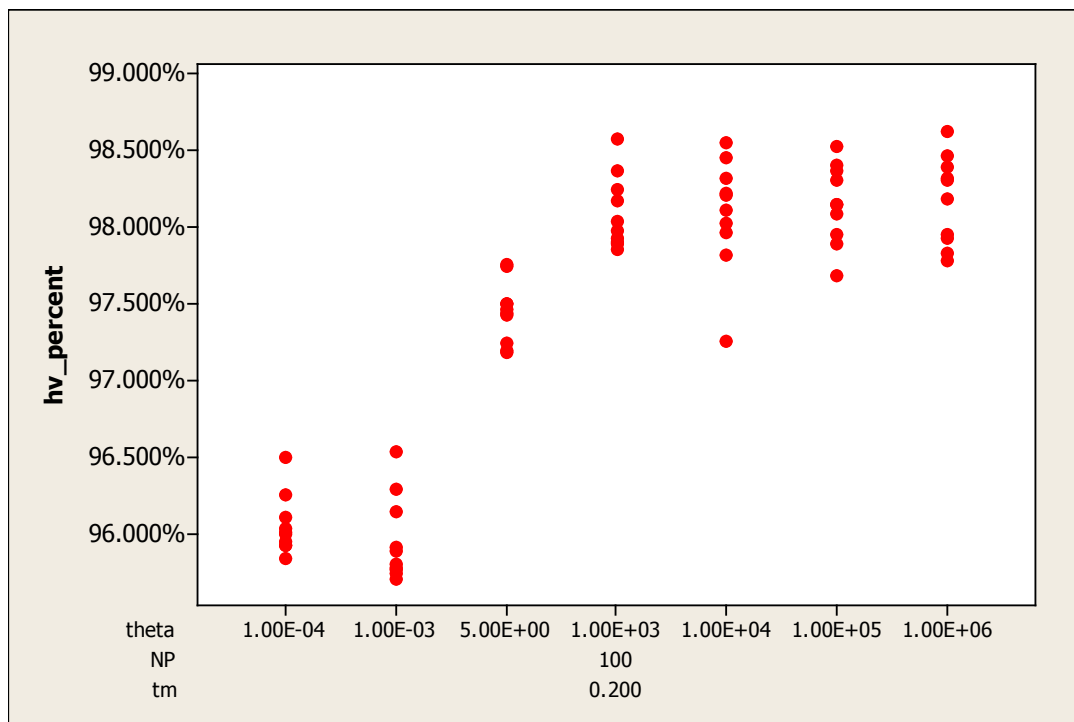
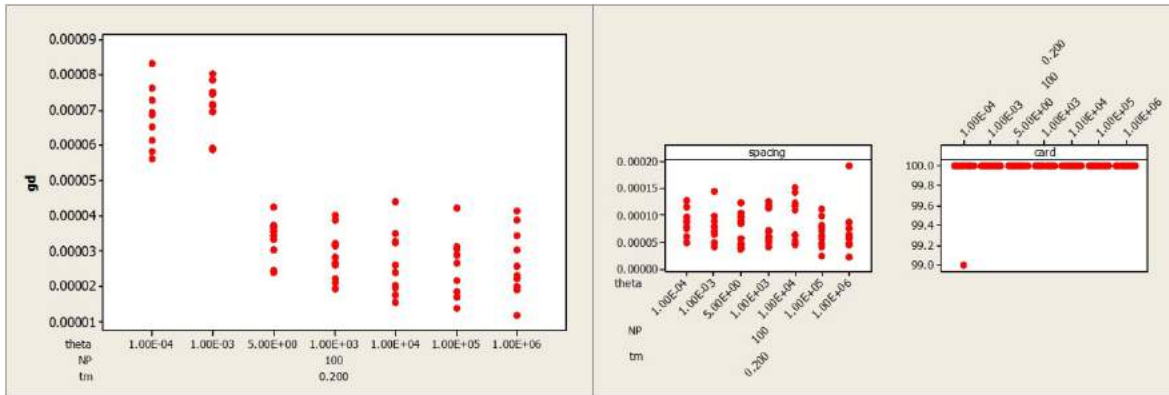


Figura 0.32 Ajuste parámetros etapa1: gd, ESS y card (MOEA/D)



Diseño factorial completo

Para la segunda etapa se probaron los valores de los parámetros de acuerdo a la Tabla 0.13.

Tabla 0.13 Valores de los parámetros etapa2 (MOEA/D)

Parámetro	Valores
Tasa de mutación (zm)	{0.01, 0.1, 0.2 }
Tamaño de la población	{20, 30, 100 , 500}
Theta (θ)	{0.1, 1, 2, 10, 30, 100, 1000 }
Tamaño del vecindario en porcentaje (T)	{20%, 10%, 13.33%}

Dos aspectos se aprecian en la Figura 0.33 y Figura 0.34: el aumento del tamaño de la población mejora la calidad de las soluciones (HV_{perc}) y aunque también lo hace con el tiempo, la tendencia no es muy pronunciada³⁹. Por otro lado los rangos de variación del tiempo son más altos con el aumento de la población y

³⁹ recuerde que en NSGA-II se requiere cada generación una clasificación de múltiples frentes de Pareto a partir de la población, mientras que en el MOEA/D se realizan a lo más $2 \cdot NP \cdot EP$ comparaciones cada generación (esto debido a que se producen dos hijos en cada cruce)

las diferencias temporales entre los resultados no son marcadas para $NP \leq 100$, se descarta entonces el valor $NP = 500$.

Figura 0.33 Ajuste parámetros etapa2: tiempo (MOEA/D)

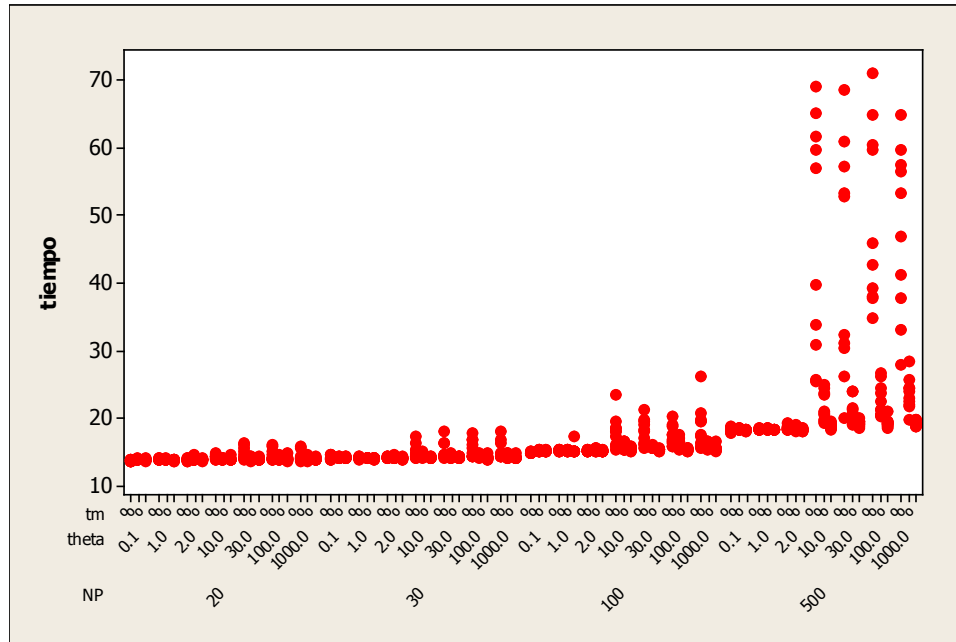
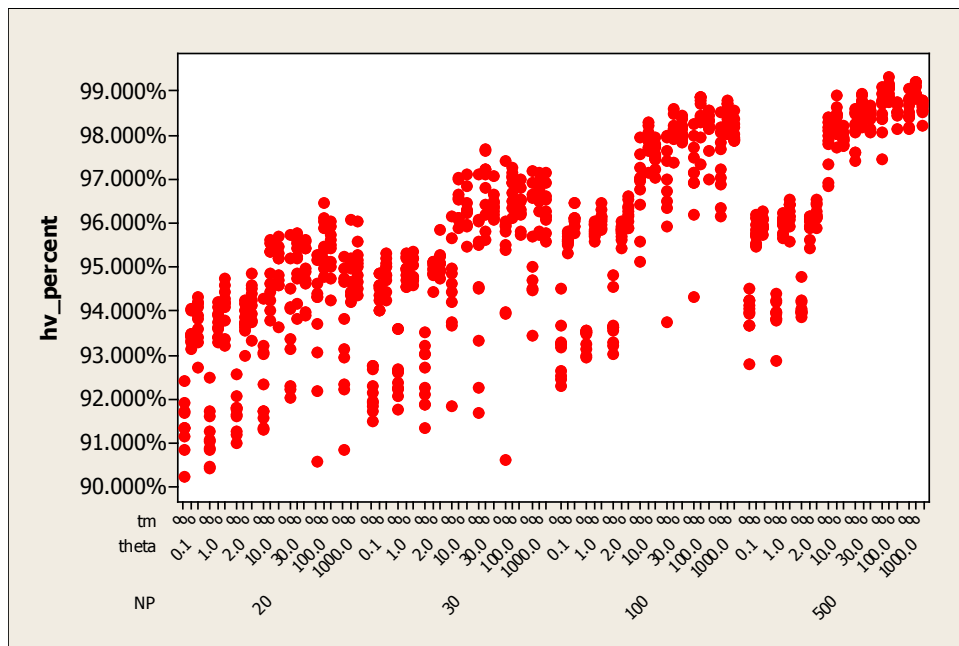


Figura 0.34 Ajuste parámetros etapa2: HV_{perc} (MOEA/D) (a)



Si la información es agrupada como se muestra en la Figura 0.35, es posible notar un incremento en la calidad de las soluciones para $\theta \geq 10$, lo cual es corroborado con la Tabla 0.14 donde. Si es definida una prueba de cola izquierda, se rechaza la hipótesis de que HV_{perc} es mejor para $\theta < 10$ que para $\theta \geq 10$, por lo cual es preciso seleccionar el parámetro θ para los valores mayores o iguales a 10. De acuerdo a la prueba en la Tabla 0.15, no es posible afirmar que el HV_{perc} difiere estadísticamente para los valores de θ restantes. Dado que hay indicios de que aumentar el valor de tal parámetro provoca aumentos en el tiempo de ejecución (Figura 0.33), se eligió $\theta = 10$.

Figura 0.35 Ajuste parámetros etapa2: HV_{perc} (MOEA/D) (b)

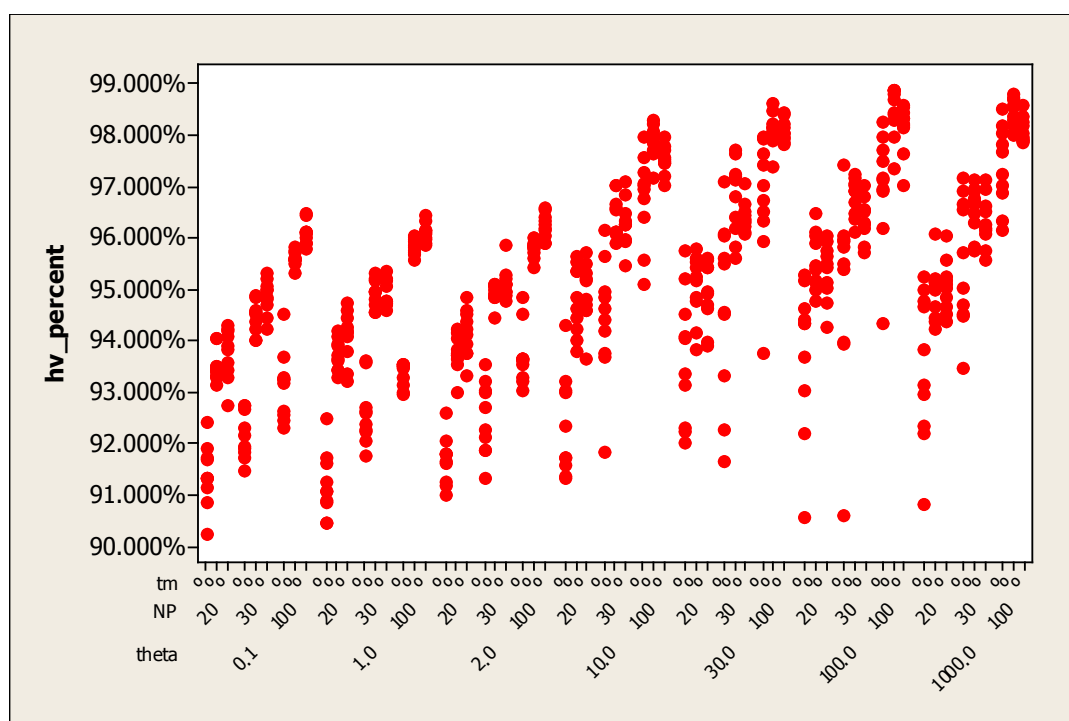


Tabla 0.14 Ajuste parámetros etapa2: Prueba de Wilcoxon (MOEA/D) (a)

Two-sample wilcoxon rank-sum (Mann-whitney) test			
theta_	obs	rank sum	expected
<10	270	54991	85185
>=10	360	143774	113580
combined	630	198765	198765
unadjusted variance	5111100.00		
adjustment for ties	0.00		
adjusted variance	5111100.00		
Ho: hv_per~t(theta_==<10) = hv_per~t(theta_==>=10)			
z = -13.356			
Prob > z = 0.0000			

Tabla 0.15 Ajuste parámetros etapa2: prueba de Kruskal-Wallis (MOEA/D)

Kruskal-wallis equality-of-populations rank test		
theta	Obs	Rank Sum
10	90	14464.00
30	90	16167.00
100	90	17444.00
1000	90	16905.00
chi-squared = 5.182 with 3 d.f.		
probability = 0.1589		
chi-squared with ties = 5.182 with 3 d.f.		
probability = 0.1589		

A continuación se elige el tamaño de la población (Figura 0.36). De acuerdo al gráfico, los mejores resultados son para $NP = 100$ usando específicamente tasas de mutación 0.1 y 0.2 (señalados con un óvalo rojo en la Figura 0.36). Realizada la prueba de cola derecha para los dos conjuntos de experimentos mencionados (Tabla 0.16) dado que se puede rechazar la hipótesis nula $H_0: HV_{perc}(NP = 100, zm = 0.10) \leq HV_{perc}(NP = 100, zm = 0.20)$ con el nivel de significancia $\alpha = 0.05$ es seleccionado $NP = 100$ con $zm = 0.10$.

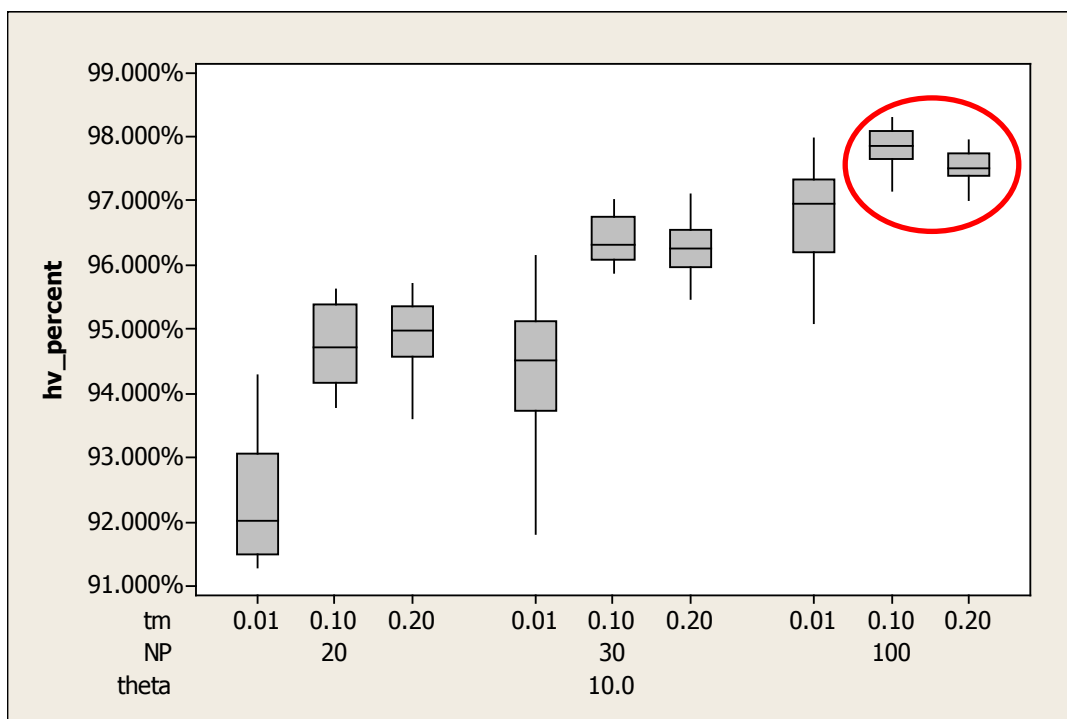
Figura 0.36 Ajuste parámetros etapa2: HV_{perc} (MOEA/D) (c)

Tabla 0.16 Ajuste parámetros etapa2: Prueba de Wilcoxon (MOEA/D) (b)

Two-sample wilcoxon rank-sum (Mann-whitney) test			
tm	obs	rank sum	expected
.10000000149	10	135	105
.20000000298	10	75	105
combined	20	210	210
unadjusted variance	175.00		
adjustment for ties	0.00		
adjusted variance	175.00		
Ho: hv_per~t(tm==.1000000014901161) = hv_per~t(tm==.2000000029802322)			
	z =	2.268	
Prob > z	=	0.0233	

Selección del tamaño de vecindario

Por último solo queda por seleccionar el valor del parámetro tamaño de vecindario (T), para ello fueron probados tres valores de la proporción que representa el vecindario con respecto al tamaño de población ($T_{perc} = \frac{T}{NP}$). Los

resultados se resumen en la Figura 0.37, y de acuerdo a la Tabla 0.17 donde se probaron los tres valores, al plantear la prueba de cola derecha en las tres pruebas se rechazan las hipótesis nulas:

$$HV_{perc}(T_{perc} = 0.10) \leq HV_{perc}(T_{perc} = 0.13)$$

$$HV_{perc}(T_{perc} = 0.10) \leq HV_{perc}(T_{perc} = 0.20)$$

$$HV_{perc}(T_{perc} = 0.13) \leq HV_{perc}(T_{perc} = 0.20)$$

con $\alpha = 0.05$. Al aceptar las hipótesis alternas como verdaderas, estadísticamente se concluye que el mejor valor es $T_{perc} = 0.1$. Dado que fue ajustado $NP = 100$, el tamaño de vecindario es $T = 10$.

En la Figura 0.38 se muestran las métricas restantes, las cuales revelan un comportamiento respecto a la calidad similar al hipervolumen.

Figura 0.37 Ajuste parámetros tamaño de vecindario: HV_{perc} (MOEA/D)

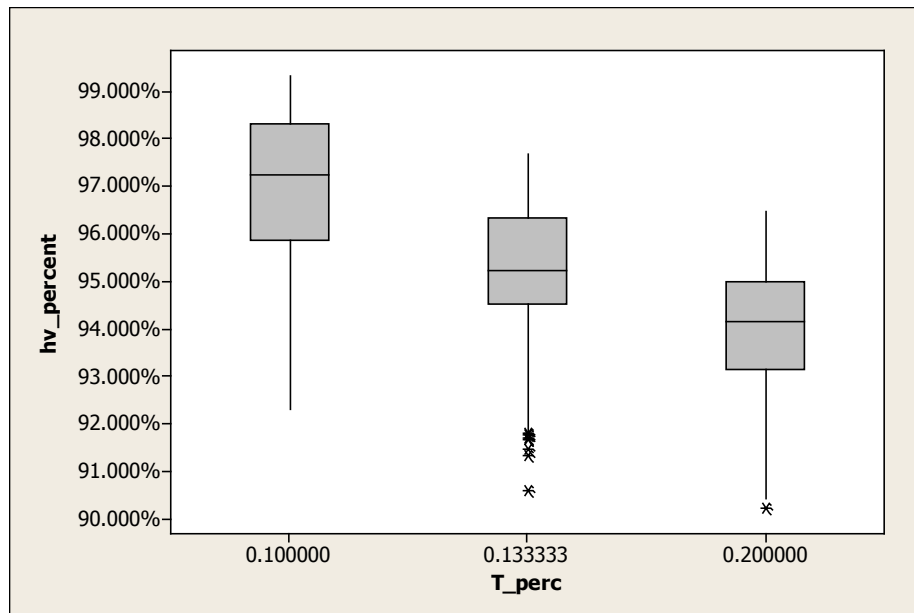
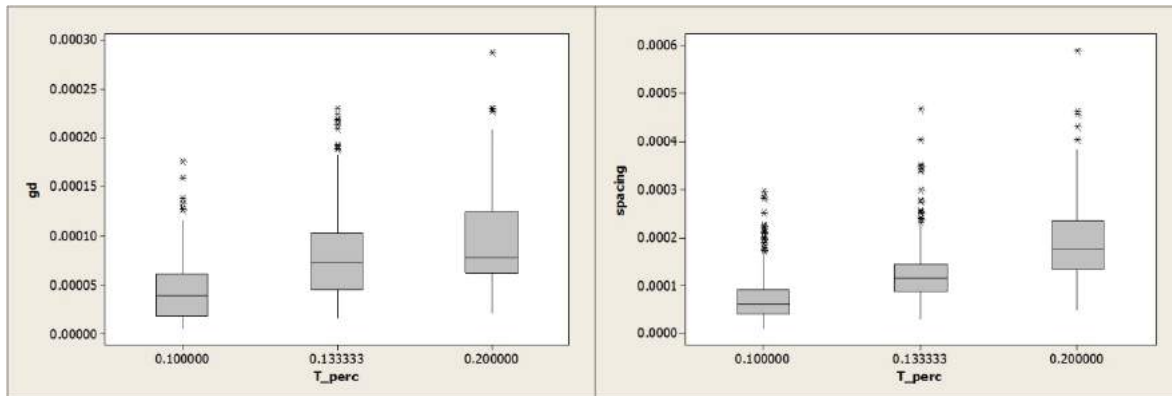


Tabla 0.17 Ajuste parámetros tamaño de vecindario: Prueba de Wilcoxon (MOEA/D)

Two-sample wilcoxon rank-sum (Mann-whitney) test				Two-sample wilcoxon rank-sum (Mann-whitney) test				Two-sample wilcoxon rank-sum (Mann-whitney) test			
t_perc	obs	rank sum	expected	t_perc	obs	rank sum	expected	t_perc	obs	rank sum	expected
10%	420	155810	132510	10%	420	167883	132510	13%	210	55030	44205
13%	210	42955	66255	20%	210	30882	66255	20%	210	33380	44205
combined	630	198765	198765	combined	630	198765	198765	combined	420	88410	88410
unadjusted variance	4637850.00			unadjusted variance	4637850.00			unadjusted variance	1547175.00		
adjustment for ties	0.00			adjustment for ties	0.00			adjustment for ties	0.00		
adjusted variance	4637850.00			adjusted variance	4637850.00			adjusted variance	1547175.00		
Ho: hv_per-t(t_perc==10%) = hv_per-t(t_perc==13%)				Ho: hv_per-t(t_perc==10%) = hv_per-t(t_perc==20%)				Ho: hv_per-t(t_perc==13%) = hv_per-t(t_perc==20%)			
z = 10.819				z = 16.425				z = 8.703			
Prob > z = 0.0000				Prob > z = 0.0000				Prob > z = 0.0000			

Figura 0.38 Ajuste parámetros tamaño de vecindario: gd y ESS (MOEA/D)



Estadísticos descriptivos de implementación final tiempo ajustado

En la sección 4.2 se reportan gráficamente los resultados definitivos de la presente investigación, en este apéndice se muestran los estadísticos descriptivos correspondientes.

Tabla 0.18 Estadísticos descriptivos de implementación final: HV_{perc}

Instancia	estrategia	media	error est. de la media	desv.est.	coef. Var.	mediana	rango intercuartil	mínimo	máximo
inst0	moead	97.53%	0.06%	0.35%	0.36%	97.60%	0.42%	96.63%	98.05%
inst0	nsga2	96.50%	0.06%	0.35%	0.36%	96.54%	0.44%	95.49%	96.98%
inst0	spo	98.31%	0.03%	0.14%	0.14%	98.29%	0.23%	98.07%	98.56%
inst0	vega	88.15%	0.44%	2.41%	2.73%	88.81%	3.19%	82.10%	92.38%
inst1	moead	95.84%	0.04%	0.21%	0.21%	95.84%	0.32%	95.38%	96.20%
inst1	nsga2	97.05%	0.05%	0.28%	0.29%	97.04%	0.41%	96.42%	97.56%

inst1	spo	97.65%	0.04%	0.24%	0.24%	97.67%	0.28%	96.78%	97.94%
inst1	vega	87.03%	0.45%	2.48%	2.85%	87.51%	2.84%	79.83%	90.84%
inst2	moead	92.78%	0.09%	0.48%	0.51%	92.76%	0.77%	91.74%	93.70%
inst2	nsga2	94.98%	0.04%	0.23%	0.25%	94.96%	0.35%	94.42%	95.39%
inst2	spo	97.22%	0.03%	0.14%	0.14%	97.22%	0.25%	96.99%	97.52%
inst2	vega	82.69%	0.42%	2.31%	2.80%	82.86%	2.73%	74.32%	85.53%
inst3	moead	95.08%	0.04%	0.19%	0.20%	95.09%	0.23%	94.70%	95.51%
inst3	nsga2	95.79%	0.06%	0.33%	0.35%	95.78%	0.42%	94.90%	96.48%
inst3	spo	98.18%	0.02%	0.10%	0.10%	98.17%	0.13%	97.98%	98.45%
inst3	Vega	81.25%	0.24%	1.32%	1.62%	81.35%	2.03%	78.92%	83.84%
inst4	moead	91.37%	0.14%	0.77%	0.84%	91.39%	1.18%	89.82%	93.11%
inst4	nsga2	93.41%	0.09%	0.47%	0.50%	93.47%	0.66%	92.30%	94.23%
inst4	Spo	98.39%	0.03%	0.17%	0.17%	98.41%	0.19%	97.99%	98.75%
inst4	Vega	83.15%	0.36%	1.95%	2.35%	83.34%	2.08%	77.90%	85.90%

Tabla 0.19 Estadísticos descriptivos de implementación final: *gd*

Instancia	estrategia	media	error est. de la media	desv.est.	coef. Var.	mediana	rango intercuartil	mínimo	máximo
inst0	moead	3.445E-05	1.773E-06	9.712E-06	28.20%	3.432E-05	1.547E-05	1.685E-05	5.460E-05
inst0	nsga2	4.833E-05	1.849E-06	1.013E-05	20.95%	4.622E-05	1.319E-05	3.422E-05	7.188E-05
inst0	Spo	2.040E-05	6.725E-07	3.684E-06	18.06%	1.993E-05	4.230E-06	1.388E-05	2.932E-05
inst0	vega	1.471E-04	4.473E-06	2.450E-05	16.65%	1.490E-04	3.543E-05	1.089E-04	1.972E-04
inst1	moead	3.636E-05	1.261E-06	6.909E-06	19.00%	3.560E-05	8.836E-06	2.412E-05	5.290E-05
inst1	nsga2	3.221E-05	9.832E-07	5.385E-06	16.72%	3.048E-05	8.730E-06	2.537E-05	4.292E-05
inst1	Spo	4.989E-06	1.405E-07	7.698E-07	15.43%	5.028E-06	1.303E-06	3.439E-06	6.350E-06
inst1	vega	2.084E-04	1.083E-05	5.929E-05	28.45%	2.031E-04	1.025E-04	1.228E-04	3.566E-04
inst2	moead	3.141E-05	9.911E-07	5.429E-06	17.28%	3.291E-05	8.055E-06	2.068E-05	4.285E-05
inst2	nsga2	2.819E-05	3.775E-07	2.068E-06	7.33%	2.839E-05	2.255E-06	2.306E-05	3.318E-05
inst2	Spo	7.772E-06	1.983E-07	1.086E-06	13.98%	7.613E-06	1.205E-06	5.972E-06	1.159E-05
inst2	vega	1.151E-04	3.266E-06	1.789E-05	15.54%	1.168E-04	2.367E-05	8.032E-05	1.578E-04
inst3	moead	7.225E-05	1.238E-06	6.781E-06	9.38%	7.115E-05	6.786E-06	5.804E-05	9.166E-05
inst3	nsga2	5.801E-05	1.354E-06	7.416E-06	12.78%	5.850E-05	9.580E-06	4.386E-05	7.758E-05
inst3	Spo	1.077E-05	1.393E-07	7.632E-07	7.08%	1.074E-05	9.068E-07	9.149E-06	1.296E-05
inst3	vega	3.261E-04	1.025E-05	5.613E-05	17.21%	3.150E-04	8.522E-05	2.289E-04	4.482E-04
inst4	moead	4.534E-05	1.809E-06	9.907E-06	21.85%	4.488E-05	1.283E-05	3.017E-05	7.558E-05
inst4	nsga2	4.206E-05	1.844E-06	1.010E-05	24.01%	4.011E-05	1.074E-05	2.340E-05	6.626E-05
inst4	Spo	2.336E-06	5.890E-08	3.226E-07	13.81%	2.279E-06	3.840E-07	1.894E-06	3.633E-06
inst4	vega	1.204E-04	3.273E-06	1.793E-05	14.89%	1.174E-04	1.978E-05	9.335E-05	1.686E-04

Tabla 0.20 Estadísticos descriptivos de implementación final: *ESS*

Instancia	estrategia	media	error est. de la media	desv.est.	coef. Var.	mediana	rango intercuartil	mínimo	máximo
inst0	moead	8.1788E-05	8.1224E-06	4.4488E-05	54.39%	5.9601E-05	5.5943E-05	4.0614E-05	0.00021408
inst0	nsga2	0.00011548	3.5993E-06	1.9714E-05	17.07%	0.00011848	2.163E-05	6.8059E-05	0.00015141
inst0	Spo	4.9889E-05	2.2966E-06	1.2579E-05	25.21%	4.6714E-05	2.0277E-05	3.557E-05	8.8532E-05
inst0	Vega	0.00031042	1.62E-05	8.8731E-05	28.58%	0.00030986	0.00013166	0.00016424	0.00050588
inst1	moead	9.7533E-05	6.9353E-06	3.7986E-05	38.95%	8.6403E-05	5.1428E-05	4.5852E-05	0.00019146
inst1	nsga2	0.0001102	4.6941E-06	2.5711E-05	23.33%	0.00010599	3.5017E-05	6.7178E-05	0.0001731
inst1	Spo	3.381E-05	1.9754E-06	1.082E-05	32.00%	3.3856E-05	2.1343E-05	1.8046E-05	4.7756E-05
inst1	Vega	0.00029923	1.5981E-05	8.7534E-05	29.25%	0.00027279	0.00010499	0.00018649	0.0005956
inst2	moead	8.0085E-05	6.4397E-06	3.5272E-05	44.04%	7.0644E-05	4.6759E-05	2.7975E-05	0.0001799
inst2	nsga2	8.3258E-05	3.9025E-06	2.1375E-05	25.67%	7.9823E-05	2.9433E-05	5.1486E-05	0.00013687
inst2	Spo	2.16E-05	8.7E-07	4.7652E-06	22.06%	1.9814E-05	5.3776E-06	1.5274E-05	3.4996E-05
inst2	Vega	0.00020047	1.2078E-05	6.6155E-05	33.00%	0.00018075	0.00010985	0.00011954	0.00037169
inst3	moead	7.8249E-05	5.7799E-06	3.1658E-05	40.46%	7.1436E-05	4.592E-05	3.5708E-05	0.00017166
inst3	nsga2	9.9577E-05	3.2114E-06	1.759E-05	17.66%	9.7052E-05	2.5922E-05	7.1203E-05	0.00014665
inst3	Spo	2.0578E-05	5.125E-07	2.8071E-06	13.64%	1.9673E-05	3.4388E-06	1.685E-05	2.7175E-05
inst3	Vega	0.00022574	1.44E-05	7.8874E-05	34.94%	0.00020874	0.00014691	0.00011338	0.00040192
inst4	moead	9.3159E-05	7.1284E-06	3.9044E-05	41.91%	8.6259E-05	7.2952E-05	4.0152E-05	0.00015827
inst4	nsga2	0.00010954	1.2832E-05	7.0282E-05	64.16%	8.0341E-05	8.338E-05	3.0512E-05	0.00029329
inst4	Spo	1.6467E-05	6.4419E-07	3.5283E-06	21.43%	1.5511E-05	2.8756E-06	1.3099E-05	2.8973E-05
inst4	Vega	0.00018175	1.5206E-05	8.3285E-05	45.82%	0.0001578	0.00013082	7.7185E-05	0.00039841

Tabla 0.21 Estadísticos descriptivos de implementación final: *card*

Instancia	estrategia	media	error est. de la media	desv.est.	coef. Var.	mediana	rango intercuartil	mínimo	máximo
inst0	moead	100	0	0	0.00%	100	0	100	100
inst0	nsga2	29.5666667	0.1037607	0.56832078	1.92%	30	1	28	30
inst0	Spo	264.466667	4.65245784	25.4825611	9.64%	266.5	39.5	211	317
inst0	vega	31.7	0.94583175	5.18053386	16.34%	32	7.25	21	41
inst1	moead	99.9666667	0.03333333	0.18257419	0.18%	100	0	99	100
inst1	nsga2	29.7333333	0.08211757	0.44977645	1.51%	30	1	29	30
inst1	Spo	998.9	8.70715784	47.6910676	4.77%	1008	82.75	887	1080
inst1	vega	21.4333333	0.68259702	3.73873788	17.44%	21.5	5.25	15	28
inst2	moead	99.7	0.15275252	0.83666003	0.84%	100	0	96	100
inst2	nsga2	29.7666667	0.07854032	0.43018307	1.45%	30	0.25	29	30
inst2	Spo	477.666667	8.11899056	44.4695427	9.31%	496.5	69	364	542
inst2	vega	18.6	0.58447343	3.20129284	17.21%	19	5.25	11	23
inst3	moead	98.9666667	0.36666667	2.00831604	2.03%	100	2	91	100

inst3	nsga2	29.6	0.09097177	0.49827288	1.68%	30	1	29	30
inst3	Spo	620.233333	7.99837388	43.808898	7.06%	625	33.25	505	709
inst3	vega	27.9333333	0.90201987	4.9405663	17.69%	30	6.5	16	35
inst4	moead	85	2.41118085	13.2065814	15.54%	80.5	25.5	62	100
inst4	nsga2	29.9	0.07350033	0.4025779	1.35%	30	0	28	30
inst4	spo	753.666667	8.65312824	47.3951353	6.29%	750.5	55.5	665	867
inst4	vega	16.9333333	0.64314839	3.52266879	20.80%	16	6.25	11	23